

Construction by Robot Swarms Using Extended Stigmergy

By Justin Werfel, Yaneer Bar-Yam & Radhika Nagpal

German translation by Johannes Irmer, University of Surrey



massachusetts institute of technology — computer science and artificial intelligence laboratory

Bau durch Roboter- schwärme unter Verwendung 5 erweiterter Stigmergie

Justin Werfel, Yaneer Bar-Yam
& Radhika Nagpal

10

AI Memo 2005-011

April 2005

15

20

25

© 20 05 massachusettsinstituteoftechnology, cambridge, ma 02139 usa — www.csail.mit.edu

30

Abstract

35

In dieser Arbeit wird ein System beschrieben, in dem einfache, untereinander identische autonome Roboter zweidimensionale Gebilde aus identischen Bausteinen errichten. Es wird gezeigt, dass es in einem System, das auf diese Weise in mobile und strukturelle Einheiten unterteilt ist, möglich ist, Robotern ein globales strukturelles Wissen zu vermitteln, das ausreicht, um in kürzester Zeit komplizierte vorentworfene Bauwerke zu bauen. Dieses Wissen wird den Robotern durch den Einsatz von Bausteinen, die über beschränkte Kommunikationsfähigkeiten verfügen, vermittelt. So wird das von sozialen Insekten verwendete Prinzip der Stigmergie (das Speichern von Informationen in der Umgebung) erweitert, indem die Fähigkeiten der Bausteine erweitert werden, die diese Umgebungsinformationen repräsentieren. Dadurch können frei wählbare feste Bauwerke mit Hilfe weniger festgelegter, lokaler Verhaltensweisen errichtet werden, ohne dass der Bau vorher genau geplant werden muss.

40

45

50

55

60

65

Diese Arbeit wurde unterstützt durch den NSF¹-Zuschuss EIA-0130391.

¹ Anm. des Übersetzers: Die NSF (National Science Foundation) ist eine amerikanische Regierungseinrichtung, die Forschung und Lehre in Wissenschaft und Technik durch Zuschüsse sowie durch Verträge und sonstige Vereinbarungen mit Universitäten, höheren Schulen, nichtkommerziellen Einrichtungen und kleineren Betrieben unterstützt. Darüber hinaus regt sie die Zusammenarbeit von Universitäten, Industrie und Regierung an und fördert die internationale Zusammenarbeit auf den Gebieten der Wissenschaft und der Technik.

1 Einleitung

70

In dieser Arbeit wird ein System des automatisierten Baus beschrieben, das aus zwei verschiedenen Funktionseinheiten besteht: Robotern (spezialisiert auf Bewegung und Manipulation) und Bausteinen (spezialisiert auf strukturelle Eigenschaften). Innerhalb dieser zwei Kategorien sind die Agenten jeweils vollkommen identisch. Es wird ein Algorithmus gezeigt, durch den Roboter dichtgepackte Bauwerke schaffen können, wenn die Bausteine vollkommen inert sind. Anschließend werden die Vorteile beschrieben, die erzielt werden, wenn man die Bausteine technisch komplexer und die Roboter technisch weniger komplex gestaltet. Besteht zwischen den Robotern keine direkte Kommunikation, reicht schon eine Gruppe einfacher, festgelegter lokaler Verhaltensweisen, um nach den Angaben des Benutzers ein beliebiges zusammenhängendes, lückenloses und zweidimensionales Bauwerk zu errichten.

75

80

85

90

Besonders in Gebieten, in denen der Einsatz menschlicher Arbeitskräfte gefährlich oder problematisch wäre, könnte die Fähigkeit des automatischen Baus von großem Nutzen sein; so könnten Roboter zum Beispiel zunächst in Unterwassergebiete oder ins Weltall geschickt werden, um dort Behausungen für den später nachkommenden Menschen zu errichten. Denkansätze, die auf Roboterschwärmen aus einer Vielzahl relativ einfacher Roboter basieren, haben bei der Verwirklichung solcher Ziele Vorteile gegenüber einem einzigen oder wenigen technisch komplizierteren Robotern, besonders was Dezentralisierung und Robustheit anbelangt. Solche Systeme können für gewöhnlich den Verlust vieler Komponenten verkraften, ohne dass die Erfüllung der Aufgabe entscheidend gefährdet wäre. Darüber hinaus müssen ihre Komponenten nicht zwangsläufig an eine bestimmte Aktionsreihenfolge gebunden sein. Aufgrund der Schwierigkeiten, das Verhalten von Robotern in unbekanntem Umgebungen bis ins Detail vorzuplanen, ist diese Eigenschaft besonders nützlich.

95

Schwärme von Robotern, die ein Gebilde errichten, erinnern natürlich unweigerlich an soziale Insekten wie Ameisen und Bienen und können von diesen lernen. Diese Insekten verwenden zur Organisation ihrer Bauaktivitäten Stigmergie: Ameisen z. B.

werden beim Ablegen von Materialien von ihrer unmittelbaren Umgebung beeinflusst, und beeinflussen diese Umgebung ihrerseits durch das Ablegen der Materialien. Auf diese Weise kommunizieren die Insekten sozusagen durch die Manipulation ihrer Umgebung.

Die Stigmergie ist ein leistungsfähiges und einfaches Werkzeug, das jedoch auch seine Grenzen hat. So kann sie zwar dazu verwendet werden, Bauwerke mit vorgegebenen Eigenschaften herzustellen [1, 11], jedoch nicht ohne weiteres zur einheitlichen Errichtung *spezifischer* (potenziell frei wählbarer und komplizierter) Bauwerke. Außerdem konnte bisher noch kein allgemeines Prinzip beschrieben werden, das es ermöglichen würde, aus einem bestimmten Bauwerk, das nachgebaut werden soll, eine Gruppe von einfachen Verhaltensweisen zu extrahieren, die es den Bauagenten erlauben würde, das Bauwerk nachzubauen. Es wird vorgeschlagen, das Baumaterial mit einer gewissen Kommunikationsfähigkeit zu versehen, und es wird gezeigt, dass die Roboter dadurch nicht länger in der Lage sein müssen, die lokale Bauwerksgeometrie zu bestimmen. Es wird außerdem gezeigt, dass die Baugeschwindigkeit sich mit zunehmender Anzahl sowohl der Roboter als auch der Bausteine erhöht, und dass das System sich auf natürliche Weise erweitert, um den Bau frei wählbarer, benutzerdefinierter finiter Formen zu ermöglichen.

In Abschnitt 2 wird ein Überblick über verwandte Arbeiten auf diesem und benachbarten Gebieten gegeben, in Abschnitt 3 werden die grundlegenden Annahmen dargelegt, auf denen diese Arbeit basiert. Abschnitt 4 enthält einen Algorithmus, den Roboter beim Umgang mit inerten Bausteinen einsetzen können. Abschnitt 5 beschreibt das Szenario, in dem kommunikative Bausteine eingesetzt werden. Abschnitt 6 vergleicht die Leistungsfähigkeit der beiden Fälle auf theoretischer Basis und auf Basis einer Simulation. Abschnitt 7 beschreibt die Fehlerkorrektur und das Abbauen der Bauwerke.

2 Verwandte Arbeiten

125 Verschiedene frühere Veröffentlichungen behandeln das Thema des Baus, wenn auch nicht
mit dem Ziel, benutzerspezifizierte Konstruktionen zu errichten. [4, 6, 13] beschäftigen sich
beispielsweise mit Aspekten des Einsatzes von Kommunikation zwischen mehreren
Robotern, um deren Effektivität bei der Erfüllung bestimmter Aufgaben zu verbessern.
[2] diskutiert Fragen des mechanischen Designs von Robotern, die aus
130 Extruderschaum Torbögen, Türme und Mauern errichten sollen. Andere Forscher
befassen sich mit der Minimalisierung der Fähigkeiten von Robotern, die Bauaufgaben
wie das Errichten von Mauern aus frisbee-ähnlichen Scheiben (sog. Pucks) [7] oder
das Räumen eines Gebiets [8] verrichten sollen. [14] beschreibt zwar ein begrenztes
Rahmenwerk zur Ordnung von Bausteinen in frei wählbare Linien und Kurven, befasst
135 sich jedoch nur mit inerten Bausteinen und benötigt daher leistungsfähigere Roboter.

Der dieser Arbeit zugrundeliegende Ansatz ist vor allem durch Forschungen auf
zwei Gebieten motiviert. Zum Einen durch Forschungen auf dem Gebiet des auf
Stigmergie basierenden Baus durch Insekten und Insekten-inspirierte Roboter, zum
Anderen durch Forschungen auf dem Gebiet der selbstkonfigurierenden modularen
140 Robotik [12, 15]. Forschungen auf dem letzten Gebiet haben äußerst leistungsfähige
Hardwaresysteme hervorgebracht, die auch für diese Arbeit eingesetzt werden. Dazu
gehören z. B. selbstabstimmende Verbindungen, durch die eine ungenaue Ausrichtung
korrigiert wird, und Kommunikationsverbindungen zwischen physisch verbundenen
Modulen. Bei Algorithmen für modulare Roboter müssen für gewöhnlich alle Module
145 ständig miteinander verbunden bleiben, und die einzelnen Module müssen über
Mobilität verfügen. Diese beiden Eigenschaften eignen sich jedoch nicht für
automatisierte Bauanwendungen. In der jüngsten Vergangenheit haben Forscher auf
dem Gebiet der modularen Robotik damit begonnen, Hardwaresysteme für die
automatisierte Errichtung von Bauwerken zu beschreiben [3, 10]. Diese Systeme
150 basieren auf inerten Modulen und haben sich bisher noch nicht damit befasst, wie das
verteilte Verhalten der Roboter bestimmt werden kann und wie komplexe

benutzerspezifizierte Bauwerke erstellt werden können. Die hier dargelegten Ergebnisse sollten auch auf diese Systeme anwendbar sein.

155 3 Grundlegende Probleme und Annahmen

Es wird der Problembereich behandelt, in dem mobile Roboter willkürlich in einem Arbeitsgebiet platziert werden, in dem an zufälligen Positionen mehrere Bausteinlager errichtet wurden. Außerdem ist das Arbeitsgebiet frei von Hindernissen und ein Marker legt den Ort fest, an dem mit dem Bau begonnen werden soll. Das Ziel der Roboter ist es, die Bausteine von den Lagern zu holen und beginnend beim Marker zu einem gewünschten Bauwerk anzuordnen (Abb. 1). Der Marker und die Lager senden eindeutig identifizierbare Kurzstreckensignale aus, die den Robotern sozusagen den Weg weisen. Als Marker dient ein Objekt mit derselben Geometrie und den selben Verbindungsmechanismen wie ein Baustein. Dieser Marker ist der „Samen“, mit dem die Bausteine als erstes verbunden werden können.

160

165

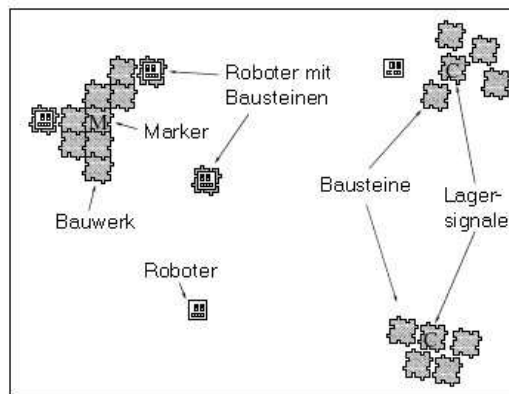


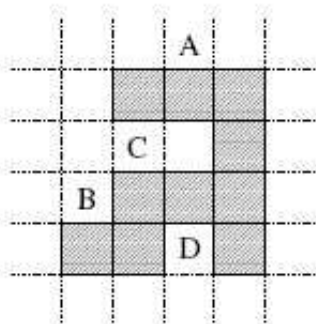
Abbildung 1: Überblick über das beschriebene System. Während des Baus bildet das Bauwerk ein Gitterwerk mit einem impliziten Koordinatensystem, als dessen Ursprung der Marker gelten kann.

170

Es wird mit rechteckigen Bausteinen gearbeitet, die in der horizontalen Ebene angeordnet werden sollen. Eine wichtige Einschränkung ist dabei, dass ein Baustein dann und nur dann zu dem wachsenden Bauwerk hinzugefügt werden kann, wenn die potenzielle Verbindungsstelle über mindestens zwei benachbarte offene Seiten verfügt.

175 Anderenfalls reicht der zum Manövrieren verfügbare Platz nicht aus, um den neuen
Baustein zu dem Bauwerk hinzuzufügen (Abb. 2). Obwohl bereits ein System
aufgezeigt wurde, in dem kubische Bausteine an freie Stellen geschoben werden
können, wie sie z. B. bei Punkt (D) in Abb. 2 auftreten [10], wird durch die Beibehaltung
dieser Einschränkung die Aufgabe der mechanischen Konstruktion erleichtert und die
180 Präzision, mit der die Roboter arbeiten müssen, verringert. Insbesondere durch die
Verhinderung von Lücken wie z. B. der Lücke am Punkt (D) werden auch Situationen
verhindert, in denen ein Roboter einen Baustein einen längeren „Tunnel“ hinab
manövrieren müsste, wie er z. B. bei Punkt (C) existiert.

185 Roboter können sich in der Ebene in jede beliebige Richtung bewegen, selbst
wenn sie einen Baustein halten. Sie können zur Kollisionsvermeidung die Anwesenheit
anderer Roboter erkennen, entweder durch aktive Kurzstreckensignale oder durch
passive Wahrnehmung. Befinden sie sich erst einmal in der unmittelbaren Umgebung
des wachsenden Bauwerks, können sie dessen Rand folgen und die Bausteine in ihrer
unmittelbaren Nachbarschaft beobachten.



190

Abbildung 2: Beispiele für gültige und ungültige mögliche Andockstellen an einem
Beispielbauwerk (schattierte Felder stehen für Gitterquadrate, die bereits mit einem Baustein
besetzt sind). An den Punkten A oder B kann jeweils ein neuer Baustein angebracht werden. Die
195 Punkte C und D sind räumlich zu eingeschränkt, um dort einen neuen Baustein in Position zu
bringen.

4 Inerte Bausteine

Einer der Hauptgründe dafür, nicht nur eine einzige Klasse mobiler Bausteine mit der Fähigkeit der Selbstkonfiguration zu verwenden, sondern die Systemelemente in mobile Roboter und passive Bausteine zu unterteilen, ist die Tatsache, dass eine Mobilität der Bauelemente nicht länger nötig ist, sobald ein Bauwerk einmal errichtet ist. Eine Optimierung der Bauelemente in Bezug auf ihre strukturellen Eigenschaften macht sie wahrscheinlich effektiver und kostengünstiger. Die extremste Umsetzung dieser Strategie ist die Verwendung von Bausteinen, die vollkommen inert sind, sich in Masse produzieren lassen und nicht komplexer als einfache Ziegelsteine sind.

4.1 Algorithmus für Roboter

Ein Aspekt des Problems ist die Vermeidung ungewollter Lücken. Dazu muss vermieden werden, dass sich zwei Bausteine in derselben Reihe befinden, die nicht direkt nebeneinander liegen. Da davon ausgegangen wird, dass die Roboter nur Zugang auf lokale Informationen haben, muss die Platzierung der Bausteine alleine auf Basis lokaler Konfigurationen bestehender Bausteine stark eingeschränkt werden, um solche Situationen zu verhindern. Eine mögliche Strategie, um das zu erreichen, ist die Errichtung des Bauwerks in Schichten. Dazu werden nach und nach neue Bausteine in einer Reihe entlang einer Kante des bestehenden Bauwerks hinzugefügt, wobei dieser Vorgang immer am selben Ende beginnt und immer in dieselbe Richtung abläuft.

Algorithmus 1 beschreibt ein Verfahren, nach dem Roboter ein dichtgepacktes Bauwerk aus inerten Bausteinen errichten können. Dieses Verfahren lässt sich durch die Regel zusammenfassen, dass ein Baustein nur an einer „rechten Ecke“ oder einer „linken Ecke“ (Abb. 3), die unmittelbar auf eine andere linke Ecke folgt, platziert werden kann. Intuitiv entspricht die Platzierung eines Bausteins an einer rechten Ecke der Platzierung am Ende einer bestehenden unvollständigen Reihe; die Platzierung an einer linken Ecke, die einer linken Ecke folgt, entspricht dem Beginn einer neuen

225 Reihe; die Platzierung an einer linken Ecke, die einer rechten Ecke folgt, würde zu zwei nicht benachbarten Bausteinen in derselben Reihe führen, was nicht erlaubt ist.

Algorithmus 1 Pseudocode-Verfahren für die Errichtung eines Bauwerks aus inerten Bausteinen durch einen einzigen Roboter.

230 **loop**

 hole Baustein von Lager

 gehe zu Rand des Bauwerks

 linke-Ecke-passiert \leftarrow false

while Baustein immer noch gehalten **do**

235 **if** an-rechter-Ecke oder (an-linker-Ecke und linke-Ecke-passiert=true) **then**

 bringe Baustein hier an

else if an-linker-Ecke **then**

 linke-Ecke-passiert \leftarrow true

 folge Rand entgegen den Uhrzeigersinn

240

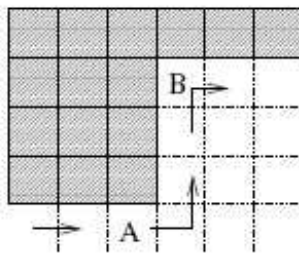
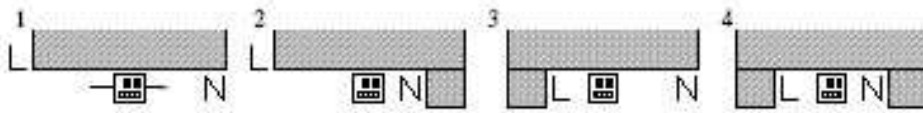


Abbildung 3: A: Eine linke Ecke. B: Eine rechte Ecke. Pfeile zeigen den Weg an, den ein dem Rand folgender Roboter zurücklegt.

245 Durch Induktion lässt sich beweisen, dass dieser Algorithmus ein dicht gepacktes Bauwerk garantiert. Man stelle sich einen Roboter vor, der dem Rand eines gültigen Bauwerks folgt (d. h. es befinden sich keine zwei Bausteine in derselben Reihe, die nicht direkt benachbart sind). Wie in Abb. 4 zu sehen, können die letzte Ecke (L) und

250 die nächste Ecke (N) auf dem Weg des Roboters jeweils entweder eine linke oder eine rechte Ecke sein:



255 Abbildung 4: Schema eines Roboters, der dem Rand des Bauwerks entgegen dem Uhrzeigersinn folgt. N und L sind die nächste bzw. die letzte Ecke auf dem Weg des Roboters. Je nach der Bauwerksgeometrie können N und L jeweils entweder eine linke oder eine rechte Ecke sein, wie hier und im Text erläutert.

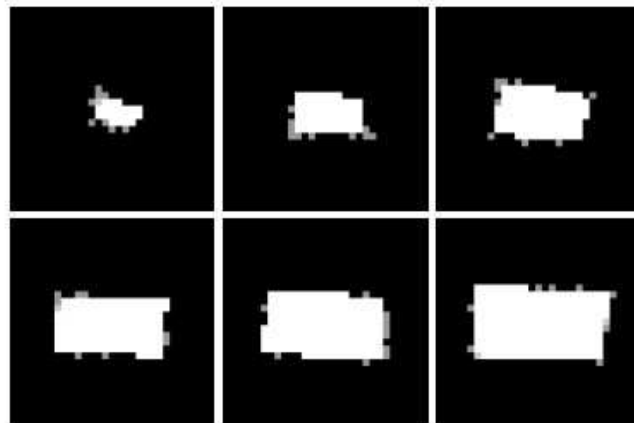
1. Wenn sowohl P als auch N linke Ecken sind, entsteht durch die Platzierung eines Bausteins bei N ein gültiges Bauwerk.
- 260 2. Wenn P eine linke Ecke und N eine rechte Ecke ist, entsteht durch die Platzierung eines Bausteins bei N ein gültiges Bauwerk.
3. Wenn P eine rechte Ecke und N eine linke Ecke ist, entsteht durch die Platzierung eines Bausteins bei N ein ungültiges Bauwerk.
- 265 4. Der Hypothese nach können P und N nicht beide rechte Ecken sein, da das bedeuten würde, dass in dieser Reihe zwei nicht benachbarte Bausteine liegen.

270 Wird den Robotern die Beschränkung einprogrammiert, Bausteine nur an rechten Ecken oder linken Ecken, die unmittelbar auf andere linke Ecken folgen, zu platzieren, lässt sich so die Gültigkeit des Bauwerks aufrechterhalten. Als letzter Punkt der Induktion ist noch zu erwähnen, dass das Bauwerk auch in seinem frühesten Stadium, in dem es nur aus einem einzigen Baustein besteht, gültig ist.

275 Abbildung 5 zeigt ein Bauwerk, das gerade mit Hilfe dieses Algorithmuses von zehn Robotern errichtet wird. Sind mehrere Roboter gleichzeitig tätig, kann es zu

Situationen kommen, in denen der Algorithmus geringfügig erweitert werden muss. So kann es zum Beispiel zu einem völligen Stillstand kommen („Gridlock“), wenn Roboter zu dicht um den Rand des Bauwerks gedrängt sind, so dass sich die Roboter weder weiter fortbewegen noch an ihrem momentanen Standort einen Baustein anbringen können. Eine Lösung ist, dass die Roboter abhängig davon, wie lange sie sich bereits nicht mehr bewegen können, einfach aufgeben und sich für kurze Zeit aus dem Umfeld des Bauwerks zurückziehen. Eine weitere Komplikation kann entstehen, wenn zwei Roboter, die sich in unmittelbarer Nähe zueinander befinden, fast im selben Moment entscheiden, einen Baustein anzubringen, der für sich alleine zwar gültig angebracht werden könnte, nicht jedoch, wenn auch der andere Baustein angebracht wird. Dieses Problem lässt sich vermeiden, indem ein Roboter, der entscheidet einen Baustein anzubringen, vorher ein Signal an alle ihm mit einem Abstand von bis zu zwei Plätzen folgenden Roboter aussendet. Dieses Signal veranlasst die übrigen Roboter dazu, ihre „linke-Ecke-passiert“-Flags zurückzusetzen.

Roboter, die in der Lage sind, zuverlässig geometrische Eigenschaften des Bauwerks zu erkennen, sind für diese Verwendung inerter Bausteine genau wie für andere auf Stigmergie basierende Anwendungen eine wichtige Voraussetzung. Diese komplizierte Aufgabe könnte z. B. durch eine in die Bausteine integrierte Schiene vereinfacht werden. Der Roboter heftet sich an diese Schiene, wenn er das Bauwerk erreicht, um sich von ihr beim Ablaufen des Randes leiten zu lassen und jede passierte Ecke ausdrücklich zu registrieren. Eine weitere Einschränkung ist die Tatsache, dass Algorithmus 1 zwar lückenlose, unendliche und flache Gebilde errichten kann, komplizierte Bauwerke in diesem einfachsten Rahmenwerk allerdings ohne z. B. eindeutige Wegweiser oder Methoden, bestimmte Bausteine markieren zu können, generell nicht möglich sind. Solche Ansätze könnten dazu dienen, die Zahl der möglichen Bauwerke, die von einem verteilten Roboterteam aus inerten Bausteinen errichtet werden können, zu erhöhen. Stattdessen ist es jedoch sinnvoll, einen Teil der Planungsarbeit auf die Bausteine selbst umzuwälzen.



305 Abbildung 5: Konstruktion eines rechteckigen Beispielbauwerks. Die Abbildung zeigt aufeinander folgende Aufnahmen, die während des Bauvorgangs gemacht wurden (von links nach rechts und oben nach unten). Weiß: Bausteine; Grau: Roboter; Hellgrau: Roboter, die dabei sind, Bausteine anzubringen; Schwarz: Leere Zellen.

310 **5 Kommunikative Bausteine**

Da die Rechenkraft von Computern heutzutage mit immer geringeren Kosten verbunden ist [5], kann den Bausteinen relativ problemlos eine gewisse Komplexität verliehen werden. Roboter sind auf Kommunikation durch externe Signale angewiesen. Diese Kommunikation kann jedoch unzuverlässig sein, durch Mehrwegeeffekte und andere Komplikationen beeinträchtigt werden und mit zunehmender Zahl der Roboter an Leistungsfähigkeit verlieren. Unter Bausteinen dagegen besteht eine direkte physikalische Verbindung, sobald sie im Bauwerk verbaut sind. Diese Verbindung kann als Basis für zuverlässige, unmissverständliche und schnelle Kommunikation dienen. Modulare Roboter verwenden solche Datenverbindungen für gewöhnlich zusätzlich zu ihren physikalischen Verbindungen.

315

320

Ein Baustein, der bereits mit dem Bauwerk verbunden ist, kann sofort von seinen Nachbarn seine Position in einem gemeinsamen Koordinatensystem abrufen. Darüber hinaus liefern die benachbarten Bausteine ihm Informationen über das aktuelle Bauwerk und die gewünschte Endform. Bausteine am Rand des Bauwerks und

325 Roboter, die sich entlang des Randes fortbewegen, können ebenfalls miteinander
kommunizieren, wenn auch mit geringerer Bandbreite. Diese Kommunikation findet auf
sehr kurzer Entfernung statt. Dadurch werden die Probleme, die bei einer
Kommunikation über größere Entfernungen hinweg auftreten können, ebenso
vermieden wie Störsignale bei gleichzeitiger Aktivität vieler Roboter und auch
330 Missverständnisse, welche Agenten gerade Signale aussenden. Sollte die oben
erwähnte Möglichkeit einer physikalischen Schiene in die Bausteine integriert werden,
könnte die Kommunikation über diese Schiene erfolgen.

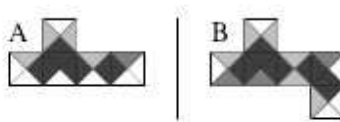
5.1 Algorithmen für Bausteine und Roboter

335 Diese Kommunikation versorgt Roboter an jedem Punkt in der Umgebung des
Bauwerks mit globalen Informationen über das Bauwerk. Roboter müssen daher keine
Aufzeichnungen über die Geometrie machen, die sie auf ihrem Weg um den Rand des
Bauwerks passiert haben. Stattdessen kann das Bauwerk selbst Plätze identifizieren,
an denen Bausteine angebracht werden können.

340 Die grundlegende Idee zur garantierten Errichtung eines dichtgepackten Bauwerks
ist die gleiche wie die zur Errichtung eines Bauwerks aus inerten Bausteinen.
Allerdings stehen nun mehr Plätze für die potenzielle Anbringung von Bausteinen zur
Verfügung. Der erste Baustein, der zu einer Reihe hinzugefügt wird, kann über die
gesamte Länge der Reihe an einem beliebigen Platz platziert werden, nicht nur am
345 Ende der Reihe. Der weitere Bau in dieser Reihe kann vom ersten Baustein aus in
beide Richtungen fortgesetzt werden, also nicht nur in eine Richtung, wie es beim Bau
mit inerten Bausteinen der Fall ist.

Algorithmus 2 beschreibt Prozeduren, die beim Bau eines dicht gepackten
Bauwerks sowohl von Robotern als auch von Bausteinen befolgt werden müssen. Die
350 Roboter umkreisen einfach das Bauwerk, bis sie einen Platz finden, der ihnen die
Anbringung eines Bausteins erlaubt. Das Bauwerk gewährleistet, dass in ein und
derselben Reihe keine zwei Bausteine angebracht werden, die nicht direkt benachbart
sind, indem jede Seite jedes Bausteins einen bestimmten Zustand zugewiesen

355 bekommt. Dieser Zustand hängt davon ab, an welchen Plätzen in der Reihe, die an die
 betreffende Seite angrenzt, bereits Bausteine angebracht sind (Abb. 6). Wurden in
 einer Reihe noch keine Bausteine angebracht, sind die entsprechenden Seiten der
 Bausteine in der benachbarten Reihe als „offen“ markiert. Wird nun ein Baustein in der
 leeren Reihe angebracht, wird durch die benachbarte Reihe eine Nachricht gesendet,
 welche die Seiten als „geschlossen“ markiert, und dadurch die Anbringung eines
 360 weiteren Bausteins in dieser Reihe verhindert. Lediglich neben bereits bestehenden
 Bausteinen, deren Seiten als „Ecke“ markiert werden, können weitere Bausteine
 angebracht werden. Seiten, die direkt an einen angebrachten Baustein angrenzen,
 werden mit dem Zustand „fertig“ markiert.



365 Abbildung 6: Zustände der Seiten der Bausteine in Algorithmus 2 werden hier als weiß (offen),
 hellgrau (Ecke), mittelgrau (geschlossen) und dunkelgrau (fertig) gezeigt. A: Zustände eines
 Beispielbauwerks. Die südlichen Seiten der Bausteine in der unteren Reihe sind „offen“, da in der
 darunter liegenden Reihe noch keine Bausteine angebracht sind. B: Wird ein Baustein in der
 370 darunter liegenden Reihe angebracht, wird die südliche Seite des benachbarten Bausteins als
 „fertig“ markiert. Die Südseite des nächsten Bausteins wird als „Ecke“ markiert, womit dort eine
 Anbringung möglich ist. Die Südseiten der übrigen Bausteine in der Reihe werden als
 „geschlossen“ markiert, womit verhindert wird, dass zwei nicht benachbarte Bausteine in
 derselben Reihe angebracht werden. Die Nordseite des neu angebrachten Bausteins wird als
 375 „fertig“ markiert (da diese Seite direkt an des Bauwerks angebracht ist), seine Ostseite wird als
 „offen“ markiert (da die Ostseite des nördlichen Bausteins „offen“ ist), seine Westseite wird als
 „Ecke“ markiert (da die Westseite des nördlichen Bausteins „fertig“ ist), und seine Südseite wird
 als „offen“ markiert (da diese Seite weder an einem anderen Baustein angebracht ist, noch
 neben einer Seite liegt, die an einem Baustein angebracht ist).

380

380 Genau wie bei Algorithmus 1 für inerte Bausteine sind gewisse Erweiterungen
notwendig, um den korrekten Betrieb zu gewährleisten, wenn mehrere Roboter zur
selben Zeit aktiv sind. Vor allem muss der Algorithmus sicherstellen, dass bei endlicher
Geschwindigkeit der Nachrichtenverbreitung innerhalb des Bauwerks nicht mehrere
385 Roboter zur gleichen Zeit die Erlaubnis erhalten können, einen Baustein in einer
„offenen“ Reihe anzubringen. Dies kann zum Beispiel gewährleistet werden, indem
entlang der Reihe eine Nachricht vorausgeschickt wird, durch die sichergestellt wird,
dass alle Bausteine in der Reihe wissen, dass die Erlaubnis zur Anbringung eines
Bausteins an einem bestimmten Platz gegeben werden soll. Bevor nun dem ersten
390 Roboter die Anbringungserlaubnis erteilt wird, wird abgewartet, bis jeder einzelne
Baustein in der Reihe bestätigt hat, dass er keinem anderen Roboter eine
Anbringungserlaubnis erteilen wird. Diese Konsultation mit der Reihe ist nur beim
ersten in einer „offenen“ Reihe angebrachten Baustein nötig. Danach nehmen die
ehemals „offenen“ Seiten in der Reihe einen anderen Zustand an und können den
395 Robotern selbständig, d. h. ohne vorherige Kommunikation mit anderen Bausteinen,
das Anbringen eines Bausteins erlauben oder verbieten.

Algorithmus 2 Pseudocode-Verfahren für den Bau eines dicht gepackten Bauwerks aus kommunikativen Bausteinen durch einen einzigen Roboter. Jede Seite jedes Bausteins hat als Zustand einen Wert aus der Menge {offen, geschlossen, Ecke, fertig}.
400 Die Funktion f wird definiert als $f(\text{fertig}) = \text{Ecke}$, $f(\text{offen}) = \text{offen}$, $f(\text{anderer Wert}) = \text{geschlossen}$.

A: *Bausteine*

```
    if Anbringung an das Bauwerk soeben abgeschlossen then
      for all Seiten S do {siehe Abb. 6}
405     if S ist direkt an Bauwerk angebracht then
          Zustand(S) ← fertig
        else if S liegt neben einer an das Bauwerk angebrachten Seite S' then
          B ← an S' angebrachter Baustein
          Zustand(S) ←  $f$ (Seite von B mit derselben Orientierung wie S)
410     else
          Zustand(S) ← offen
        else if bereits Teil des Bauwerks then
          if Roboter fragt, ob ein Baustein an Seite S angebracht werden kann then
            if Zustand(S) = geschlossen then
415             antworte Nein
            else if Zustand(S) = Ecke then
                antworte Ja
            else {Zustand(S) ist offen}
                antworte Ja
420             sende Nachricht „schlieÙe(S)“ an benachbarte Bausteine beiderseits von S
                {diese Nachricht verbreitet sich in der gesamten Reihe}
          if erhalte Nachricht „schlieÙe(S)“ then
              Zustand(S) ← geschlossen
```

425 sende Nachricht „schlieÙe(S)“ an Baustein auf gegenüberliegender Seite von
 sendendem Baustein

if ein neuer Baustein soeben an Seite S angebracht wurde **then**

 Zustand(S) ← fertig

 sende Nachricht „(S) angebracht“ an benachbarte Bausteine beiderseits von S

if erhalte Nachricht „(S) angebracht“ **then**

430 Zustand(S) ← Ecke

 {markiere neu entstandene rechte Ecken, um Anbringung zu erlauben}

B: Roboter

while Bauwerk nicht abgeschlossen **do**

 hole Baustein von Lager

435 gehe zu Rand des Bauwerks

while Baustein immer noch gehalten **do**

 frage umliegende Bausteine, ob Baustein hier angebracht werden kann

if alle Bausteine des Bauwerks Ja antworten **then**

 bringe Baustein hier an

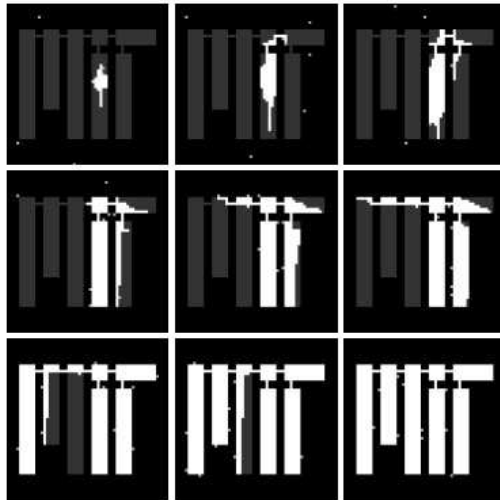
440 **else**

 folge Rand entgegen den Uhrzeigersinn

5.2 Zusammengesetzte Bauwerke

445 Anders als bei inerten Bausteinen lässt sich der Algorithmus problemlos auf Bauwerke mit begrenzten Ausmaßen und frei wählbarer Form ausweiten. Interne Lücken sind noch immer verboten und Freiräume zwischen Bausteinen im gewünschten Endbauwerk müssen mindestens zwei Gitterzellen breit sein, damit die Roboter weiterhin ungehindert dem Rand des Bauwerks folgen können. Eine gegenüber Algorithmus 2 nahezu unveränderte Prozedur erlaubt absichtliche Einstülpungen des
450 Randes des Bauwerks. Der Unterschied liegt darin, dass nun unter der Voraussetzung, dass sie durch einen Raum getrennt sind, der im Endbauwerk frei bleiben soll, zwei Bausteine in derselben Reihe angebracht werden können, ohne dass sie nebeneinander liegen müssen.

Dank dieser Änderung kann der Benutzer ganz einfach ein Bauwerk mit jeder
455 beliebigen Form spezifizieren, solange es die oben genannten Kriterien erfüllt. Die Form kann als binäre Belegungsmatrix, oder noch kompakter als Ansammlung von Rechtecken dargestellt werden, deren Superposition das gewünschte Bauwerk ergibt [9]. Diese Information kann vor dem Bau gemeinsam mit dem Marker gespeichert werden, und danach entlang eines gemeinsamen Koordinatensystems an
460 nachfolgende Bausteine weitergegeben werden, wenn diese dem Bauwerk hinzugefügt werden. Abb. 7 zeigt ein Beispiel eines Bauwerks in Form des MIT-Logos (wobei die Buchstaben miteinander verbunden sind, um ein einziges zusammenhängendes Bauwerk zu schaffen). Der Bauvorgang kann sich in seinen Details, d. h. darin, welcher Teil des Bauwerks zu welcher Zeit errichtet wird, in individuellen Läufen unterscheiden,
465 die Endform entspricht jedoch zuverlässig dem im Voraus festgelegten Bauwerk.



470

Abbildung 7: Beispielkonstruktion eines zusammengesetzten Bauwerks. Aufeinander folgende Aufnahmen, die während des Bauvorgangs gemacht wurden (von links nach rechts und oben nach unten). Weiß: Bausteine; Grau: Roboter; Dunkelgrau: Bereiche, in denen einmal Bausteine platziert werden; Schwarz: Bereiche, in denen nie Bausteine platziert werden.

6 Vergleich der Leistungsfähigkeit

475

Im Folgenden wird der Bau mit inerten Bausteinen mit dem Bau mit kommunikativen Bausteinen unter Verwendung der zwei Algorithmen in Bezug auf die benötigte Bauzeit verglichen. Es werden theoretische und experimentelle Ergebnisse zum Zeitverhalten präsentiert, sowohl was (A) die Anzahl der Bausteine im Bauwerk und (B) die Anzahl der Roboter angeht. Simulationen finden auf einem Gitter statt. Eine Zelle kann leer sein, oder einen Baustein, einen Roboter ohne Baustein oder einen Roboter, der einen Baustein trägt, enthalten.

480

6.1 Auswirkungen der Größe des Bauwerks

485

Zunächst wird die Frage untersucht, wie die für den Bau eines Bauwerks benötigte Zeit sich im Verhältnis zu ihrer Größe verändert. In diesem Abschnitt werden die zwei Extreme eines einzelnen Roboters und einer unbeschränkten Anzahl an Robotern behandelt.

Im Allgemeinen wird die Zeit, die benötigt wird, um ein Bauwerk aus n Bausteinen zu errichten, eine Funktion aus verschiedenen Faktoren sein: N , der Zahl der Roboter, die am Bau mitwirken; L , der Zeit, die nötig ist, um einen Baustein vom Lager zu holen und ihn zum Rand des Bauwerks zu transportieren; D , der durchschnittlichen Entfernung, die ein Roboter entlang des Randes des Bauwerks zurücklegen muss, bevor er einen Platz findet, an dem er einen Baustein anbringen kann; A , der Zeit, die nötig ist, um einen Baustein am Bauwerk anzubringen. Als Zeiteinheit wird die Zeit verwendet, die nötig ist, um die Länge einer Zelle des Gitters zurückzulegen.

Definitionsgemäß benötigt ein einzelner Roboter durchschnittlich die Zeit $L + D + A$, um einen Baustein zu holen und anzubringen. L ist abhängig von Faktoren wie den Standorten der Lager im Arbeitsgebiet, A ist abhängig von der mechanischen Konstruktion der Hardwareimplementierung; D dagegen ist hauptsächlich vom Algorithmus abhängig. Eine Worst-Case-Analyse der relativen Veränderungen von $D(n)$ kann sowohl für inerte als auch für kommunikative Bausteine durchgeführt werden. Interessanter ist jedoch der typische Fall, der sich einfacher durch Simulation erforschen lässt.

$D(n)$ wird im Experiment gemessen, indem wiederholt ein einziger Roboter von einem zufällig ausgewählten Ort entlang eines großen Kreises, dessen Mittelpunkt der Marker ist, in Richtung des Markers geschickt wird. Hat der Roboter den Rand des Bauwerks erreicht, wird aufgezeichnet, wie viele Zellen er zurücklegen muss, bis er einen gültigen Platz zum Anbringen eines Bausteines gefunden hat. Bei inerten Bausteinen wächst D in etwa linear mit n (Abb. 8). Dies spiegelt die Tatsache wider, dass Bauwerke, die auf diese Weise errichtet werden, meist stark in die Länge gezogen sind. Bei einem quadratischen Bauwerk müsste ein Roboter für gewöhnlich eine Strecke von etwa \sqrt{n} zurücklegen, um einen gültigen Platz zum Anbringen eines Bausteins zu finden. Bei kommunikativen Bausteinen wächst D etwa mit $n^{0.6}$. Bei Einsatz eines einzigen Roboters beträgt die für den Bau eines Bauwerks aus n inerten

Bausteinen daher $O(n^2+n(L+A))$, für den Bau eines Bauwerks aus kommunikativen

515

Bausteinen $O(n^{1.6}+n(L+A))$.

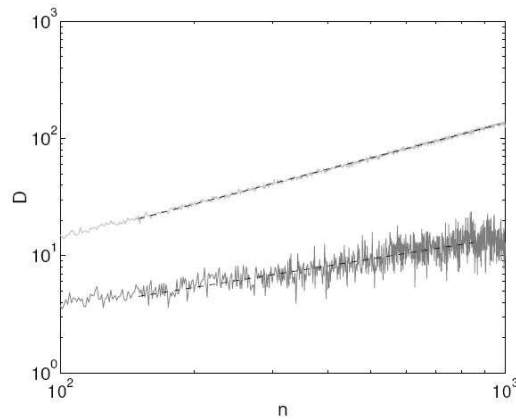


Abbildung 8: Durchschnittliche Entfernung D , die ein Roboter, der das Bauwerk erreicht, zurücklegen muss, um einen Platz zu finden, an dem er seinen Baustein anbringen kann, als Funktion der Anzahl der Bausteine n , die bereits Teil des Bauwerks sind. Durchschnittswert aus 100 Läufen. Hellgrau: Inerte Bausteine; Dunkelgrau: Kommunikative Bausteine; gestrichelte Linien geben die kleinsten Fehlerquadrate auf dieser doppelt-logarithmischen Skala an. Es ergibt sich $D(n) \sim O(n^{0,994 \pm 0,002})$ für inerte Bausteine, $D(n) \sim O(n^{0,61 \pm 0,02})$ für kommunikative Bausteine.

520

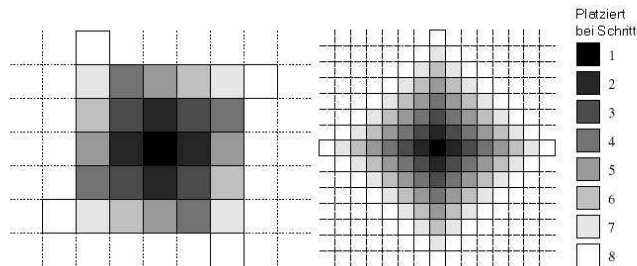
525

Werden Störungen zwischen Robotern ignoriert, gibt es in dem Grenzbereich, in dem N sehr groß ist, immer einen Roboter, der verfügbar ist, um einen Platz mit einem Baustein zu besetzen, sobald die Genehmigung dazu erteilt wird. Abb. 9 zeigt, dass es unabhängig von der Größe des Bauwerks bei inerten Bausteinen bei jedem Schritt vier genehmigte Plätze gibt, da die Roboter eine linke Ecke passieren müssen, bevor sie einen Baustein an einer linken Ecke anbringen. Die Zeit, die nötig ist, um n Bausteine anzubringen, beträgt in diesem Fall $O(nA)$. Bei kommunikativen Bausteinen vergrößert sich die Anzahl der genehmigten Plätze linear mit der Nummer des Schritts (zusammen mit dem Rand des Bauwerks). Bei Schritt s werden also $O(s)$ Bausteine angebracht. Dies ergibt eine kumulative Gesamtzahl von $O(s^2)$ und die Zeit, die nötig

530

535 ist, um ein Bauwerk aus n inerten Bausteinen zu errichten, ändert sich entsprechend zu $O(\sqrt{n} A)$.

Tabelle 1 fasst diese Ergebnisse zusammen.



540 Abbildung 9: Bei einer beliebig großen Anzahl an Robotern N kann am Ende jedes Schrittes der Länge A davon ausgegangen werden, dass an jedem Platz, an dem dies erlaubt ist, Bausteine platziert werden. In diesen Diagrammen werden Bausteine, die in aufeinanderfolgenden Schritten angebracht werden, in zunehmend helleren Schattierungen angezeigt. Links: Inerte Bausteine; Rechts: Kommunikative Bausteine.

545

	Inerte Bausteine	Kommunikative Bausteine
$t(n) (1)$	$O(n(L+D+A))$	$O(n(L+D+A))$
$t(n) (2)$	$O(nA)$	$O(\sqrt{n}A)$
$D(n) (1)$	$O(n)$	$O(n^{0.6})$

Tabelle 1: Zusammengefasste Ergebnisse aus §6.1. (1) $N = 1$; (2) N beliebig groß.

6.2 Auswirkungen der Roboteranzahl

550 Als nächstes wird untersucht, wie die für die Errichtung eines Bauwerks mit festgelegter Größe nötige Zeit sich im Verhältnis zu der Anzahl an Robotern verändert. Zu diesem Zweck wird der Raum zwischen den Grenzwerten $N = 1$ und $N \rightarrow \infty$ untersucht. Dazu werden im Experiment N Roboter simuliert, die ein Quadrat mit einer

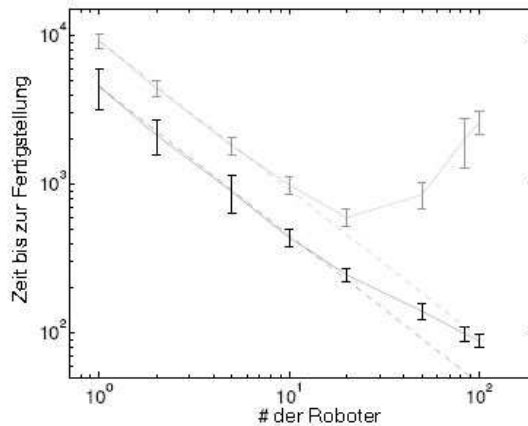
Seitenlänge von 21 Bausteinen errichten¹. In diesen Simulationen gilt $L = 0$ und $A = 1$.

555 Wie in den $D(n)$ -Experimenten aus §6.1 nähern die Roboter sich von einem Kreis aus dem Bauwerk. Wird ein Roboter von einem anderen Roboter blockiert, der sich vor ihm befindet, wartet er, bis der erste Roboter sich wieder in Bewegung setzt, bevor er sich selbst weiterbewegt.

560 Abb. 10 zeigt, wie sich die für die Fertigstellung des Bauwerks benötigte Zeit im Durchschnitt von 100 unabhängigen Läufen mit der Anzahl der Roboter N verändert, sowohl für inerte als auch für kommunikative Bausteine². Bei Verwendung von kommunikativen Bausteinen ist der Bau sehr viel schneller abgeschlossen als beim Einsatz von inerten Bausteinen, im Durchschnitt um einen Faktor von mehr als 2. Dies war durchgehend zu beobachten, ganz gleich wie groß die getestete Roboteranzahl
565 war. In beiden Fällen können bis zu 10 Roboter gleichzeitig an einem Bauwerk dieser Größe arbeiten, ohne dass es zu bedeutenden Störeffekten kommt – N Roboter schließen den Bau des Bauwerks in $1/N$ der Zeit ab, die ein einzelner Roboter dafür benötigt.

¹ Roboter, die Algorithmus 1 für inerte Bausteine verwenden, würden über die festgelegte Größe hinaus weiterbauen. Zu Vergleichszwecken wird hier verhindert, dass Bausteine außerhalb des vorgegebenen Quadrats angebracht werden.

² Bei insgesamt fünf Läufen, bei denen inerte Bausteine und 84 bzw. 100 Roboter eingesetzt wurden, war der Gridlock so extrem, dass über einen Zeitraum von 12000 Zeiteinheiten kein Baustein angebracht werden konnte.



570 Abbildung 10: Durchschnittliche Zeit bis zur Fertigstellung für ein quadratisches Bauwerk mit
 einer Seitenlänge von 21 Bausteinen als Funktion der Anzahl der Roboter. Graue Linien stehen
 für inerte Bausteine, schwarze für kommunikative Bausteine. Die gestrichelten Linien zeigen, wie
 schnell der Bau fertig gestellt wäre, wenn es keine Störungen gibt und N Roboter $1/N$ der Zeit
 benötigen, die ein einzelner Roboter benötigt.

575

Ab 20 Robotern treten Störeffekte auf. Bei inerten Bausteinen steigt die für die
 Fertigstellung des Baus benötigte Zeit parallel zu N an. Roboter müssen das Bauwerk
 selbst untersuchen können, um zu bestimmen, ob eine linke Ecke ein gültiger Platz
 zum Anbringen eines Bausteins ist, anstatt sich einzig auf lokale Informationen
 verlassen zu können. So kann es zum Gridlock kommen, wenn $N \gtrsim$ der Länge des

580

Randes des Bauwerks ist, wodurch die Roboter sich weder bewegen noch an ihrem
 aktuellen Standort Bausteine anbringen können. Bei kommunikativen Bausteinen
 nimmt die absolute für die Fertigstellung benötigte Zeit weiter ab, wenn die Zahl der
 Roboter vergrößert wird, wenn auch weniger schnell. Wenn das Bauwerk Robotern
 sofort mitteilen kann, ob an einem Platz ein Baustein angebracht werden kann oder
 nicht, kann die Gesamtzahl der momentan verfügbaren gültigen Plätze ein limitierender
 Faktor sein, es kommt jedoch unter den Bedingungen dieser Simulation nicht zum
 Gridlock.

585

Abb. 11 zeigt für inerte und kommunikative Bausteine sowie für 5 und 50 Roboter
 die Anzahl der Plätze, an denen während des Bauverlaufs theoretisch Bausteine

590

angebracht werden könnten. Bei kommunikativen Bausteinen stehen erheblich mehr Plätze gleichzeitig zum Bau zur Verfügung als bei inerten Bausteinen. Bei kommunikativen Bausteinen erhöht sich die maximal mögliche Zahl gültiger Plätze, wenn die Anzahl der Roboter erhöht wird. Das System macht sich die zusätzlichen Roboter zunutze, so dass der Bau gleichzeitig an mehreren Punkten entlang des Randes fortgeführt werden kann (genau wie bei der großen Veränderung von N in §6.1). Bei inerten Bausteinen dagegen sind immer nur wenige Plätze gleichzeitig verfügbar, an denen Bausteine angebracht werden können. Wird die Anzahl der Roboter erhöht, führt dies nur zu Staus, die den Bauvorgang aufhalten, ohne dass sich dabei die Zahl der Plätze entlang des Randes vergrößern würde, an denen gleichzeitig Bausteine angebracht werden können.

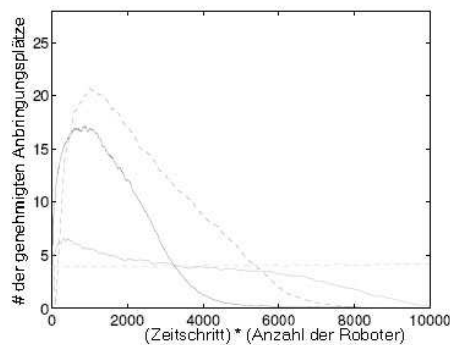


Abbildung 11: Durchschnittliche Anzahl der Plätze, an denen Bausteine angebracht werden können. Werte für kommunikative (schwarz) oder inerte Bausteine (grau), bei Einsatz von 5 (durchgezogene Linie) oder 50 (gestrichelte Linie) Robotern. Die Zeitachse bezieht die Anzahl der Roboter mit ein, so dass alle Kurven in einer einzigen Darstellung erscheinen. Die Kurven erreichen und überschreiten ihren Höhepunkt, wenn die Grenzen der Baufläche von 21 x 21 Feldern erreicht sind und das Wachstum des Bauwerks eingeschränkt wird.

Es werden keine Störungen durch eine hohe Anzahl von Robotern an anderen Stellen im Arbeitsgebiet simuliert. Dafür gibt es zwei einfache Gründe: Erstens ist es das Ziel dieser Arbeit, die Verwendung der Algorithmen und ihre Bedeutung für das Verhalten von Robotern im Umfeld des Bauwerks zu untersuchen. Zweitens sind die Roboter für gewöhnlich am Rand des Bauwerks am dichtesten gedrängt, so dass

615 eventuelle Störungen, wenn überhaupt, dann in der unmittelbaren Umgebung des
Bauwerks auftreten.

7 Zusätzliche Arbeitsvorgänge

620 Es wird kurz darauf eingegangen, wie zwei verwandte Arbeitsvorgänge, der Abbau und
die Fehlerkorrektur, mit solch einem System bewerkstelligt werden könnten. In Hinblick
auf die Fähigkeiten, die von den Robotern beherrscht werden müssen, lassen sich
beide Vorgänge mit kommunikativen Bausteinen einfacher bewältigen als mit inerten
Bausteinen.

625 7.1 Abbau

Viele Anwendungen werden Bauwerke umfassen, die nur vorübergehend benötigt
werden. Der modulare Aufbau dieses Systems ist besonders für solche Fälle geeignet,
da das Bauwerk nach der Benutzung wieder abgebaut werden kann und seine Bauteile
zur Errichtung eines neuen Bauwerks verwendet werden können. Der Abbau kann
630 durchgeführt werden, indem die Roboter dem Rand des Bauwerks folgen und
sämtliche Bausteine entfernen, welche die folgenden Kriterien erfüllen:

1. Der Baustein hat höchstens zwei Nachbarn, die an nebeneinander liegende
Seiten angrenzen müssen, d. h. wenn dieser Platz nicht besetzt wäre, könnte
635 dort ein Baustein angebracht werden (siehe Abb. 2).

2. Hat der Baustein zwei Nachbarn, die an nebeneinander liegende Seiten
angrenzen, ist der andere Platz, an den diese beiden Nachbarn angrenzen, mit
einem Baustein besetzt (siehe Abb. 12).

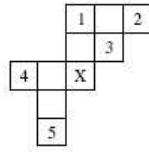
640 3. Der Baustein ist nicht am Marker angebracht, es sei denn, er ist der letzte
verbleibende Baustein. Der am Marker angebrachte Baustein muss als letztes
entfernt werden.

645 Durch das zweite und das dritte Kriterium wird sichergestellt, dass das Bauwerk
während des gesamten Abbaus aus einem zusammenhängenden Stück besteht, das
ein Roboter, der auf den Marker ausgerichtet ist, auch finden kann. Wird eine dieser
Regeln verletzt, könnte es passieren, dass ein Teil des verbleibenden Bauwerks oder
sogar das gesamte Bauwerk isoliert wird, und die Roboter es nicht mehr erreichen
650 können, indem sie einfach dem Markersignal zum Bauwerk folgen und anschließend
den Rand des Bauwerks ablaufen. Ist das Bauwerk über den ersten angebrachten
Baustein fest mit einem Ankerpunkt verbunden (wie es z. B. bei Einsätzen im Weltraum
vermutlich der Fall wäre), würde eine Verletzung dieser Kriterien bedeuten, dass die
Bausteine wegdriften und buchstäblich verloren gehen könnten.

655
Beim Einsatz mehrerer Roboter muss auch darauf geachtet werden, dass nicht zwei
Bausteine gleichzeitig gewählt werden, von denen jeder Baustein einzeln zwar die drei
Kriterien erfüllt, Kriterium 2 durch die Entfernung des jeweils anderen Bausteins aber
verletzt werden würde. Die Bausteine 1 und 3 in Abb. 12 sind ein Beispiel für ein
660 solches Paar.

Bei inerten Bausteinen müssen die Roboter daher für alle acht Zellen, die den zu
entfernenden Baustein umgeben, feststellen können, ob diese Zelle besetzt oder nicht
besetzt ist. Bei kommunikativen Bausteinen kann diese Information von den
Bausteinen selbst bereitgestellt werden.

665



665

Abbildung 12: Ein Bauwerk im Abbau. Nummerierte Bausteine können entfernt werden. Der mit X markierte Baustein erfüllt das im Text erläuterte Kriterium 1, nicht aber Kriterium 2. Würde er entfernt, würde das Bauwerk dadurch in zwei Teile geteilt werden. Der Marker kann sich an einem beliebigen der unmarkierten Plätze befinden.

670

7.2 Fehlerkorrektur

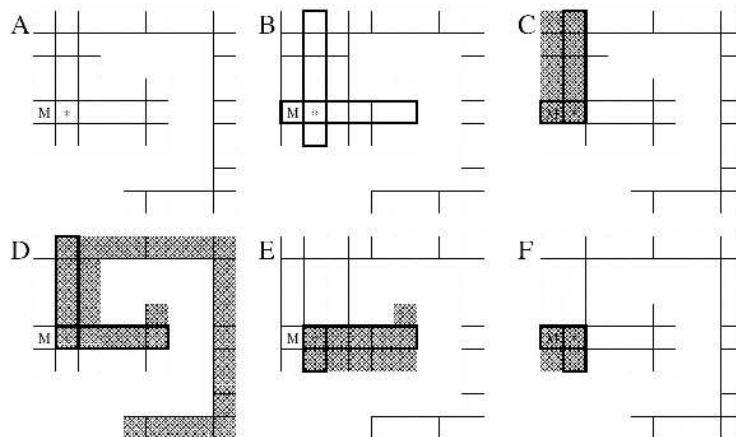
Man nehme an, dass ein Baustein versehentlich an einer falschen Stelle angebracht wurde oder aus irgendeinem Grund ausgetauscht werden muss. Es kann eine gewisse Zeit dauern, bis dieser Fehler auftritt oder entdeckt wird. Es wird daher nun das Problem des Austauschs eines beliebigen Bausteins des Bauwerks betrachtet. Beschrieben wird der Ansatz auf höchster Ebene; von der Erläuterung eines Algorithmuses, der von individuellen Bausteinen ausgeführt wird, wird abgesehen.

675

Aufgrund der Annahme, dass ein Baustein nur dann entfernt werden kann, wenn er bereits auf zwei Seiten frei liegt, müssen alle Bausteine in einem gesamten „Quadranten“ entfernt werden, um den Zielbaustein freizulegen. Abb. 13 zeigt die Abgrenzung der Quadranten. (A) zeigt ein fertig gestelltes zusammengesetztes Bauwerk. Der Marker befindet sich bei M, der zu ersetzende Baustein bei *. Durch das Ziehen horizontaler und vertikaler Arme vom Zielbaustein ausgehend (B) werden die vier (überlappenden) Quadranten definiert, die in (C)-(F) schattiert sind. Im Prinzip entsteht ein Quadrant durch Überflutung des Gebiets zu einer Seite eines Armpaars, die Arme selbst mit eingeschlossen.

680

685



690 Abbildung 13: Fehlerkorrektur in einem zusammengesetzten Bauwerk. Der Marker befindet
sich bei M, der zu ersetzende Baustein bei *. Erläuterungen im Text.

Man beachte, dass ein Quadrant bei zusammengesetzten Bauwerken Bausteine
enthalten kann, deren absolute Koordinaten über die der jeweiligen Arme hinausgehen
können (wie z. B. in (D), wo ein Quadrant, der eigentlich über dem rechten Arm liegt,
Bausteine mit einschließt, die sich weiter unten befinden). Darüber hinaus müssen
695 ebenfalls alle Bausteine zu diesem Quadranten gezählt werden, die bei Entfernung
eines Quadranten vom Rest des Bauwerks isoliert blieben. Dieses Problem tritt auf,
wenn ein Arm teilweise oder über seine gesamte Länge hinweg an einer Kante des
Bauwerks liegt. In diesem Beispiel liegt der rechte Arm an der Oberseite des zweiten
und dritten Bausteins neben dem Zielbaustein an einer solchen Kante. Alle Bausteine,
700 die weiter rechts und über diesem Arm liegen, müssen daher Teil des Quadranten sein,
der eigentlich darunter liegt, wie in (E) zu sehen ist.

Um die Fehlerkorrektur möglichst effektiv ablaufen zu lassen, sollte der Quadrant
mit den wenigsten Bausteinen entfernt werden. Genau wie beim Abbau muss auch hier
705 der Baustein, der mit dem Marker verbunden ist, bestehen bleiben, so dass alle
Quadranten, in denen der Marker enthalten ist, nicht in Frage kommen. In diesem
Beispiel enthält der Quadrant in (F) die wenigsten Bausteine, kann jedoch nicht entfernt

werden, da er den Marker enthält (genau wie der Quadrant in (C)). Es wird daher der Quadrant in (E) ausgewählt.

710 Bei kommunikativen Bausteinen werden die Roboter von den Bausteinen im
gewählten Quadranten angewiesen, keine weiteren Bausteine an sie anzubringen und
die Bausteine im Zielquadranten abzubauen. Einige Roboter ändern ihre Aufgabe bei
Erhalt einer solchen Anweisung vorübergehend von „Bau“ zu „Abbau“. Wurde der
Zielbaustein entfernt, kann dieser Abschnitt des Bauwerks durch den normalen
715 Bauvorgang erneut errichtet werden. Während dieses gesamten Korrekturvorgangs
kann der Bau in anderen Teilen des Bauwerks wie gewohnt fortgesetzt werden. Inerte
Bausteine bieten keine derartig geradlinige Möglichkeit zur Implementierung einer
Fehlerkorrektur.

720 8 **Schlussfolgerungen**

In dieser Arbeit wurde ein frühes System zum automatisierten Bau durch einen
Schwarm von Agenten umrissen und es wurden einfache Algorithmen beschrieben,
durch die Roboter lückenlose Bauwerke aus Bausteinen errichten könnten. Es wurde
bewiesen, dass eine aktive Form der Stigmergie, bei der in der Umgebung
725 gespeicherte Informationen auch zwischen Objekten in dieser Umgebung ausgetauscht
werden können, in verschiedener Hinsicht Vorteile gegenüber einer passiven Form hat
(in der die Objekte vollkommen inert sind). Roboter müssen die geometrischen
Eigenschaften des Bauwerks nicht erkennen können. Der Bau geht in absoluter
Hinsicht sehr viel schneller voran, und die relative Zeit, die benötigt wird, um ein
730 gewünschtes Bauwerk zu errichten, verringert sich mit zunehmender Größe des
Bauwerks. Der Parallelismus des Schwarms kann besser ausgenutzt werden, da der
Bau an einer Vielzahl von Punkten in der Umgebung des Bauwerks gleichzeitig
ablaufen kann. Lähmende Störungen durch den Einsatz vieler Roboter werden
drastisch verringert. Auf einfache und natürliche Weise lassen sich nun auch Bauwerke

735 mit frei wählbaren Umrissen errichten, ohne dass dafür kompliziertere Fähigkeiten oder Verhaltensweisen seitens der Roboter nötig wären.

Da einzelnen Robotern keine bestimmten zentralen Aufgaben zugewiesen werden und die Algorithmen nicht davon abhängen, dass Aufgaben in einer bestimmten Reihenfolge erfüllt werden, ist das System dem vorübergehenden oder dauerhaften
740 Ausfall von Robotern gegenüber kaum mehr anfällig, solange eine gewisse Anzahl an Robotern einsatzfähig bleibt. Ebenso wenig hängt die Erfüllung der Aufgabe von einzelnen Bausteinen ab. Es ist zwar nicht wünschenswert, dass kommunikative Bausteine versagen, nachdem sie mit dem Bauwerk verbunden wurden, doch durch eine Prozedur zur Fehlerkorrektur kann das System auch diese Eventualität
745 verarbeiten. Der Marker und die Lagersignale sind mögliche Schwachstellen des vorliegenden Systems, da das System unter Umständen die ihm gestellte Aufgabe nicht erfüllen kann, wenn sie versagen. Eine Möglichkeit, sich gegen einen Ausfall des Markers oder der Lagersignale abzusichern wäre, die Roboter oder Bausteine mit der Fähigkeit zu versehen, diese zu ersetzen [14].

750 Verschiedene Systeme zum automatisierten Bau, die inerte (aber spezialisierte) Baumaterialien verwenden [3, 10], befinden sich zur Zeit in der Frühphase ihrer Entwicklung. Die Ergebnisse der vorliegenden Arbeit lassen darauf schließen, dass es von großem Nutzen sein könnte, diese Materialien mit der Fähigkeit der Kommunikation auszustatten.

755 Themen zukünftiger Arbeiten werden heterogene Bausteine als Baumaterial, eine Erweiterung der Algorithmen auf Bauwerke mit internen Lücken sowie der Bau in drei Dimensionen sein.

Danksagung

760 Wir danken Daniela Russ und Jason Redi für nützliche Gespräche.

Literatur

- [1] Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press Inc.
- 765 [2] Bowyer, A. (2000). *Automated Construction using Co-operating Biomimetic Robots* {Technical Report, University of Bath Department of Mechanical Engineering}. Bath, UK.
- [3] Everist, J., et al. (2004). A system for in-space assembly. In *Proc. IROS 2004*,
770 Sendai, Japan: 2356-2361.
- [4] Gerkey, B. & Mataric, M. (2002). Pusher-watcher: an approach to fault-tolerant tightly-coupled robot coordination. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, D.C., USA: 464-469.
- 775 [5] Hill, J., et al. (2000). System architecture directions for networked sensors. In *ASPLOS-IX*, Cambridge, MA.
- [6] Jones, C. & Mataric, M. (2004). Automatic synthesis of communication-based
780 coordinated multi-robot systems. In *Proc. IROS 2004*, Sendai, Japan: 381-387.
- [7] Melhuish, C., Welsby, J. & Edwards, C. (1999). Using templates for defensive wall building with autonomous mobile ant-like robots. In *Proc. Towards Intelligent Autonomous Mobile Robots 99*, Manchester, UK.
- 785 [8] Parker, C., Zhang, H. & Kube, R. (2003). Blind bulldozing: multiple robot nest construction. In *Proc. IROS 2003*, Las Vegas, USA.

- 790 [9] Støy, K. & Nagpal, R. (2004). Self-reconfiguration using directed growth. In *Proc. DARS 2004*, Toulouse, France.
- [10] Terada, Y. & Murata, S. (2004). Automatic assembly system for a large-scale modular structure: Hardware design of module and assembler robot. In *Proc. IROS 2004*, Sendai, Japan: 2349-2355.
- 795 [11] Theraulaz, G. & Bonabeau, E. (1995). Coordination in distributed building. *Science* **269**: 686-688.
- [12] Vona, M. & Rus, D. (2001). Crystalline robots: self-reconfiguration with compressible unit modules. *Autonomous Robots* **10**(1): 107-124.
- 800 [13] Wawerla, J., Sukhatme, G. & Matarić, M. (2002). Collective construction with multiple robots. In *Proc. 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland.
- 805 [14] Werfel, J. (2004). Building blocks for multi-agent construction. In *Proc. DARS 2004*, Toulouse, France.
- [15] Yim, M., Zhang, Y., & Duff, D. (2002). Modular robots. *IEEE Spectrum* **39**(2):
- 810 30-34

