

Ameisenalgorithmus zur Optimierung von Modellparametern in der Modellierung komplexer Materialflusssimulationsmodelle

Ant Algorithm for the Optimization of Model Parameters during Material Flow Modelling of Complex Simulation Models

Christoph Laroque, Sebastian Krimmer
Heinz Nixdorf Institut der Universität Paderborn, Paderborn (Germany)
christoph.laroque@hni.uni-paderborn.de, skrimmer@mail.uni-paderborn.de

Abstract: This article describes an automatic approach for a specific modelling support within material flow simulation studies. Based on the adoption of classical algorithms from the artificial intelligence, especially ant algorithms, parameters at forks as well as joins within material flow models are adjusted automatically. The designed method is implemented within the material flow simulation tool d³FACT insight, developed at the Heinz Nixdorf Institute of the University of Paderborn. First results show a good possibility for application, since even initial worst-case parameter settings are automatically arranged to a more optimal solution.

1 Motivation

Die heutige Angebotsstruktur in vielen Bereichen der industriellen Fertigung ist gekennzeichnet durch verkürzte Produktlebenszyklen, kundenorientierte Produktion und eine erhöhte Variantenvielfalt. Um dennoch kosten- und zeiteffizient fertigen zu können, werden neue Produkte fast vollständig am Rechner konstruiert, modelliert und optimiert. Die Vorteile dieser Methodik liegen unter anderem in reduzierten Entwicklungskosten und -zeiten (VDI 2008). Die fortschreitende Digitalisierung geht über reine Produktmodelle weit hinaus und bezieht sich zunehmend auf die Planung, Inbetriebnahme und den Betrieb aller relevanten Prozesse der Leistungserstellung (VDI 2008). Eine etablierte Methode zur Planung, Absicherung und Verbesserung von Produktionsprozessen ist die Materialflusssimulation. Typische Probleme, die in dieser Domäne untersucht werden, sind Planungsabsicherungen, Losgrößenplanung, Variantenplanungen von Layouts, Puffergrößenbestimmung, Personaleinsatzplanung und insbesondere auch die Entwicklung und Anpassung von Steuerungsregeln im zu planenden System (LAW, KELTON 2000).

Der Simulationsexperte steht innerhalb dieser Entwicklung insbesondere bei komplexen Simulationsmodellen (z.B. der Planung von Gepäckförderanlagen in Flug-

hären usw.) vor der Fragestellung, welche Parameter für die jeweiligen Steuerungsregeln an einer bestimmten Materialflussverzweigung oder Zusammenführung verwendet werden sollen, um beispielsweise den Durchsatz des Gesamtsystems unter den gegebenen, zumeist noch stochastischen Einflüssen (Verfügbarkeiten von Maschinen, schwankende Bearbeitungszeiten, Ausfallraten usw.) zu verbessern. Hier unterstützt die hier vorgestellte Arbeit den Anwender im Rahmen der Modellierungsphase durch Verwendung eines automatischen Verfahrens bei einer möglichst guten Einstellung eben dieser Parameter. Es basiert auf einer Metaheuristik aus dem Bereich der künstlichen Intelligenz, dem Ameisenalgorithmus (DORIGO u.a. 1991).

2 Materialflusssimulation mit d³FACT insight

Am Heinz Nixdorf Institut wird seit Jahren der Materialflusssimulator d³FACT insight entwickelt, der neue Einsatzbereiche der Ablaufsimulation erschließen und den Anwender besser bei der Durchführung von Simulationsstudien unterstützen soll (LAROQUE, 2007; DANGELMAIER, LAROQUE 2008). Zurzeit befindet sich d³FACT insight in einer Redesign-Phase, wird aber stetig weiterentwickelt. Aktuell werden alle Funktionen für die Materialflusssimulation als eine Java-Programm-bibliothek (API) zur Verfügung gestellt, in der Simulationsmodelle modelliert, simuliert, analysiert und visualisiert werden können (d3FACT10-ol).

Die durch die Modellierung und anschließende Simulation erzielbare Performance eines modellierten Systems entscheidet häufig über die Auswahl einer bestimmten Planungsvariante und damit der Ergebnisse auf die Realität. Gerade im Hinblick auf die Modellierung der einzelnen Systemalternativen stellt sich also die Frage, inwiefern selbst erfahrene Simulationsexperten systematisch die entstehenden Lösungen von Hand verbessern können. Insbesondere bei stochastischen Einflüssen im Modell tritt hier in der realen Anwendung vor allem das Problem der Anzahl benötigter Simulationsläufe auf. Mit der Evaluierung und Verfeinerung jeder Konfiguration ist ein mehr oder weniger großer Rechen- und Zeitaufwand verbunden, weil durch die stochastischen Einflüsse für jedes Szenario mehrere Simulationsläufe benötigt werden (LAW, KELTON, 2000). Eine Optimierung der Modellparameter einer bestimmten Konfiguration findet in der betrieblichen Anwendung zumeist durch iterative Modellmodifikation statt. Im Hinblick auf die Zielsetzung des Beitrags stellt sich die zentrale Frage: Können Methoden der künstlichen Intelligenz aus dem Bereich des Operations Research (OR) eine automatisierte Optimierung eines konkreten Materialflussmodells hinsichtlich eines Maximierungsziels erzielen?

3 Vorbild Natur

Für viele der heute im Bereich des Operations Research (OR) verwendeten, heuristischen Lösungsverfahren stand die Natur in der ein oder anderen Form Pate. So lässt sich beispielsweise folgende Beobachtung machen: Ameisen bilden auf der Suche nach Futter Straßen zwischen ihrem Nest und der entsprechenden Futterquelle. Erstaunlich ist die Beobachtung, dass diese sich selbst organisierenden Ameisenstraßen scheinbar immer die direkte Verbindung finden, also den kürzesten Weg bilden. In einem sog. Double-Bridge-Experiment konnten Biologen bestätigen, dass Ameisen im Rahmen ihrer Futtersuche immer den kürzeren Weg finden (GOSS u.a. 1989). Dazu nutzen sie zwei wesentliche Eigenschaften:

1. Ameisen markieren während der Futtersuche ihren Weg mit Pheromonen (chemischer Duftstoff). Diesen können ihre Artgenossen wahrnehmen.
2. Die Ameisen treffen bei ihrer Wegentscheidung eine Wahrscheinlichkeitsauswahl. Je stärker der Weg mit Pheromonen markiert ist, desto wahrscheinlicher ist die Auswahl dieses Wegs für eine konkrete Ameise im Rahmen der Futterbeschaffung.

Der Italiener Marco Dorigo war der erste, der dieses Schema der Futtersuche bei Ameisen auf ein klassisches Optimierungsproblem, das Traveling Salesman Problem (TSP) übertrug (DORIGO 1991). Weitere Arbeiten haben mehrfach gezeigt, dass solche "Ameisenalgorithmen" auf verschiedene Arten von kombinatorischen Optimierungsproblemen angewendet werden können. Um den Ameisenalgorithmus möglichst flexibel auf Optimierungsprobleme anwenden zu können, existieren in der Literatur Erweiterungen und Verbesserungen (DORIGO, STÜTZLE 2004).

4 Idee: Der Anti-Ant-Algorithmus

Im Rahmen der hier vorgestellten Arbeit soll das Prinzip des Ameisenalgorithmus dazu verwendet werden, den maximalen Durchsatz durch ein vorhandenes Materialflussmodell zu identifizieren, indem die einzelnen Steuerungsparameter an Materialflussverzweigungen und Zusammenführungen so angepasst werden, dass der Gesamtfluss durch das Materialflussmodell maximiert wird. Eine simulative Evaluation der gefunden Konfiguration(en) soll die Auswahl des Ameisenalgorithmus in einem nachfolgenden Schritt validieren bzw. evaluieren. Der klassische Ameisenalgorithmus beantwortet die Frage, welchen Weg eine Ameise in einem Modell gehen muss, um als schnellstes im Ziel zu sein. Setzt man analog zu einer Ameise hier ein Werkstück im Materialflussmodell, würde dieses mit Hilfe der Ameisenalgorithmen also sukzessive den schnellsten Weg durch das Materialflussmodell finden. Da in den klassischen Ansätzen der Ameisenalgorithmen die Beschränkung des Durchsatzes (Kapazitäten von Maschinen usw.) einzelner Wege nicht berücksichtigt wird, sind diese in ihrer ursprünglichen Form nicht anwendbar. Innerhalb der hier beschriebenen Arbeiten wurde der klassische Algorithmus dementsprechend so modifiziert, dass er nun die Frage beantworten kann, wie viele Werkstücke (resp. Ameisen) maximal in einem Zeitintervall im Modell im Ziel ankommen können. Dies entspricht der eigentlichen Aufgabenstellung nach der Suche des maximalen Durchsatzes.

Im Gegensatz zum klassischen Ameisenalgorithmus erhöht der hier entwickelte "Anti-Ant-Algorithmus" die Pheromon-Konzentration auf denjenigen logischen Verknüpfungen zwischen den Bausteinen des Materialflussmodells, auf denen die Ameise nicht gelaufen ist. Sofern keine dieses Vorgehen überschreibenden Regeln im Modell definiert sind, entspricht dieser Ansatz der Gleichverteilung zunächst dem Anspruch, möglichst viel Durchsatz zu erreichen.

Für die Entwicklung des Anti-Ants wurde dazu zunächst ein einfaches Materialflussmodell gewählt (siehe Abb. 1). Die Verteilung der abgebildeten Weiche soll in diesem einfachen Beispiel von dem entwickelten Anti-Ant-Algorithmus so eingestellt werden, dass möglichst viele Werkstücke im Ziel ankommen. Der Anti-Ant-Algorithmus wird dabei jeweils aufgerufen, sobald eine Säge ein Werkstück fertig

bearbeitet hat. Im weiteren Verlauf wird dadurch sichergestellt, dass sich eine "gewichtete Verteilung" auf Basis der Leistung des der Weiche nachfolgenden Materialflusses abbilden lässt bzw. aus dem Algorithmus resultiert. Wie die spätere Analyse zeigt, können damit beispielsweise unterschiedliche Maschinenkapazitäten oder -leistungen berücksichtigt werden.

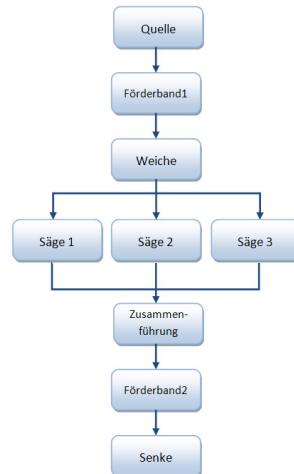


Abbildung 1: Beispielmodell zur Parameteroptimierung

5 Implementierung

In dem gewählten Beispielmodell führt von der Weiche eine logische Verknüpfung zur jeder Säge (zunächst sind hier alle Maschinen gleich). Je höher die Pheromon-Konzentration auf dieser Kante des Materialflussmodells ist, desto wahrscheinlicher wird ein Werkstück diesen Weg wählen. Jede Säge besitzt ein Attribut "letzte Bearbeitungszeit", das jeweils der letzten benötigten Bearbeitungszeit eines Werkstücks entspricht. Wenn nun ein Werkstück von einer der anderen Sägen fertig bearbeitet wurde, wird der entwickelte Anti-Ant-Algorithmus aufgerufen (vgl. Abb. 2). Anhand der Bearbeitungszeit des aufrufenden Werkstücks werden nun die Pheromon-Konzentrationen auf allen übrigen Wegen zeitgewichtet angepasst, mit Ausnahme desjenigen Weges, den das bearbeitete Werkstück passiert hat.

Die Pheromon-Konzentration, die auf einen Weg addiert wird, wird berechnet durch das Verhältnis von der Bearbeitungszeit des Werkstücks und der letzten Bearbeitungszeit der Säge zu der der Weg führt. Implizite Annahme ist hier, dass die ankommenden Werkstücke prinzipiell auf allen Sägen des Modells weiterverarbeitet werden können. Programmausdruck 1 zeigt die grundsätzliche Strategie des Anti-Ant-Algorithmus in Pseudocode. In dieser einfachen Implementierung des Anti-Ant-Algorithmus sind sinnvolle Erweiterungen, beispielsweise die Verdunstung von Pheromonen über die Zeit, bisher noch nicht integriert.

Der Zeitpunkt zu dem Säge i ein Werkstück von der Weiche zur Bearbeitung erhält, wird im Array *pst* gespeichert. Sobald das Werkstück fertig bearbeitet wurde, ruft Säge i den Anti-Ant-Algorithmus auf. Wie in Zeile 1 zu sehen ist, wird die letzte Bearbeitungszeit, die der Zeitdauer entspricht, die die Säge i für dieses Werkstück

benötigt hat, in das Array *pt* eingetragen. Das Array *fork* speichert die Pheromon-Konzentration für jeden Weg als Doublezahl. *fork(f)* enthält zum Beispiel Pheromon-Konzentration für die Säge *f*. Beim ersten Aufruf des Algorithmus wurden für die anderen Sägen noch keine letzten Bearbeitungszeiten erfasst, daher werden diese in Zeile 4 geschätzt, indem die letzte Bearbeitungszeit von Säge *i* für alle anderen Sägen angenommen wird. Die Schleife in Zeile 2 berechnet nun für alle anderen Sägen (*f* ≠ *i*) neue Pheromon-Konzentrationen, in dem das Verhältnis der letzten Bearbeitungszeiten von Säge *i* und Säge *f* auf die Pheromon-Konzentration von *f* addiert wird. Als letztes werden die neu berechneten Pheromon-Konzentrationen der Weiche in Zeile 7 zugewiesen.

```

anti_ant(i) {
1. pt(i) = now() - pst(i)
2. for(f = 0; f < fork.length;
f++) {
3.     if (pt(f) == 0) {
4.         pt(f) = pt(i)
5.     }
6.     if (f != i) {
7.         fork(f) = fork(f) +
(pt(i) / pt(f))
8.     }
9. }
10. updateforkallocation(fork)

```

Abbildung 2: Pseudocode des Anti-Ant-Algorithmus

Im Beispiel-Model sind 3 Sägen vorhanden. Ein Werkstück wird von der Weiche mit folgender Wahrscheinlichkeit zu Säge *i* weitergeleitet:

$$P(i) = \frac{fork(i)}{\sum_{f=1}^n fork(f)} \quad (1)$$

Solange Säge *i* noch ein Werkstück in Bearbeitung hat, blockiert die Weiche, welche ein Werkstück an Säge *i* weiter reichen will, diese Verknüpfung und reicht das Werkstück an freie Sägen weiter.

6 TestszENARIO

Im ersten Testmodel wurden die Verarbeitungsgeschwindigkeiten der einzelnen Stellen wie folgt in Tabelle 1 dargestellt gewählt.

Weiche und Zusammenführung verbrauchen keine Zeit. Ein Werkstück kann somit nach 13 Zeiteinheiten die Senke erreichen. Da alle Sägen hier zunächst gleich

schnell arbeiten ist die optimale Verteilung der Weiche $P(i) = \frac{1}{n}$, d.h. in einem

Drittel der Fälle leitet die Weiche ein Werkstück an Säge i weiter, wenn diese optimal eingestellt ist. Das Model wurde dreimal für jeweils 10000 Zeiteinheiten simuliert. Im ersten Simulationslauf wurde die Verteilung der Weiche dabei so gewählt, dass nur Säge 1 Werkstücke von der Weiche erhält (worst-case).

Stelle	Geschwindigkeit
Quelle	1
Förderband1	1
Säge1	10
Säge2	10
Säge3	10
Förderband2	1

Tabelle 1: Konfiguration der Verarbeitungsgeschwindigkeiten im Testmodell

Hintergrund war die Fragestellung, inwiefern durch den entwickelten Ameisenalgorithmus nicht nur eine optimale Einstellung ermöglicht werden kann, sondern auch, ob basierend auf schlechten Startwerten eine evolutionäre Entwicklung hin zu einer "optimalen" Einstellung der Bausteine ermöglicht wird. Ausgehend von der wenig optimalen Verteilung des ersten Simulationslaufs, wurde im zweiten Simulationslauf die Verteilung mit Hilfe des Anti-Ants-Algorithmus im Zeitverlauf verbessert. Um zu zeigen, wie effizient der Anti-Ant-Algorithmus sich dem maximalen Durchsatz des Modells nähert, wurde im dritten Simulationslauf die Weiche von Anfang an mit der optimalen Verteilung betrieben, um in der grafischen Auswertung den entsprechenden Benchmark darzustellen. Die resultierende Abbildung 3 zeigt die fertigen Werkstücke pro 100 Zeiteinheiten der einzelnen Simulationsläufe.

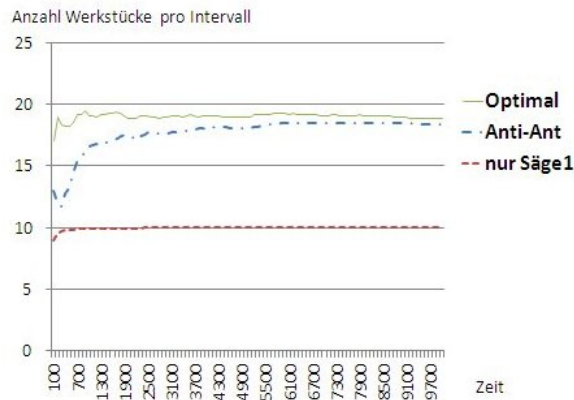


Abbildung 3: Auswertung des Testmodells 1 unter Nutzung des Ameisenalgorithmus

Im zweiten, angepassten Testmodell wurden die Bearbeitungszeiten der Sägen unterschiedlich gewählt. Auch hier wurde das Model im Rahmen des eigentlichen Simulationsexperimentes jeweils dreimal für 10000 Zeiteinheiten simuliert. Der

erste Simulationslauf wurde erneut mit einer schlechten Verteilung initialisiert, indem alle Werkstücke an Säge 1 weitergereicht wurden. Im zweiten Simulationslauf wurde die schlechte Verteilung aus dem ersten Simulationslauf unter Anwendung des Anti-Ant-Algorithmus verbessert. Der dritte Simulationslauf lief mit der für dieses Modell noch leicht berechenbaren, optimalen Verteilung.

Hier entspricht die optimale Wahrscheinlichkeit für Säge i der 'Anzahl Werkstücke pro Zeiteinheit von Säge i ' geteilt durch die 'Summe der Werkstücke pro Zeiteinheit von allen Sägen f ':

$$P(i) = \frac{wze(i)}{\sum_{f=1}^n wze(f)} \quad (2)$$

Die erzielten Ergebnisse zeigen, dass der Anti-Ant-Algorithmus funktioniert und klassische Restriktionen in Materialflussmodellen bereits in dieser einfachen Implementierung berücksichtigt werden. Sukzessive nähert sich der Algorithmus automatisch der optimalen Einstellung an (vgl. Abb. 4).

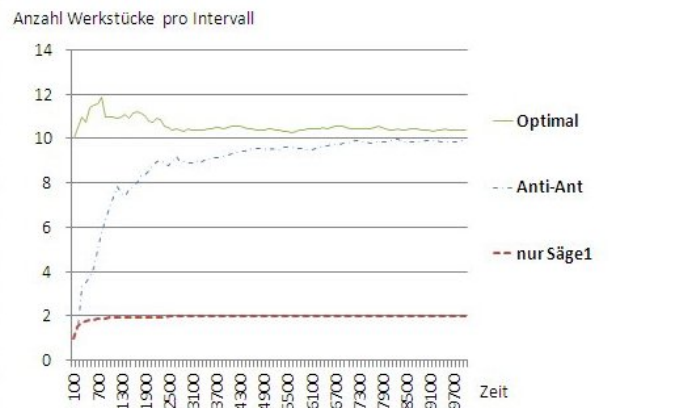


Abbildung 4: Auswertung des Testmodells 2 (versch. Kapazitäten)

So kann eine beliebig vorgegebene Verteilung eines Modells schnell und abhängig von den anderen Modellparametern optimiert werden. Die Implementierung des Algorithmus, bzw. der benötigten Bausteine im Materialflussmodell ist bewusst flexibel gehalten. Zusammengehörige Abschnitte können durch den Anwender einfach definiert und für die Berechnung der letzten Bearbeitungszeit verwendet werden. Somit können nicht nur einzelne Maschinen, sondern die Auslastung ganzer Materialflusketten definiert und optimiert werden. Überdies ist es durch die Struktur des verwendeten Materialflusssimulators möglich, die Anzahl der Sägen oder deren Bearbeitungszeiten in einem Model interaktiv zu verändern und automatisch über die Anpassung die optimale Weichenverteilung neu zu errechnen.

7 Fazit und Ausblick

Dieser Artikel beschreibt den Entwurf eines Verfahrens auf Basis der Prinzipien der Ameisenalgorithmen, um Simulationsexperten im Rahmen der Modellierung von Simulationsmodellen zur Materialflusssimulation bei der Einstellung von Materialflussmodellen zu unterstützen. Basierend auf der Annahme, dass die Maximierung des Durchsatzes ein erwünschtes Ziel der Planung ist, zeigen die erste Implementierung des Ansatzes und die Auswertung eine gute Funktionsweise. Dabei spielt die initiale Einstellung der Modellparameter beim Start der Verbesserung eine untergeordnete Rolle, kann durch den Simulationsexperten also eher glücklich oder unglücklich gewählt werden. Im einfachsten Fall kann hier also automatisch eine Gleichverteilung modelliert werden, die Verbesserung der Parametrierung erfolgt dann automatisch. Sukzessive werden über die Anwendung des vorgestellten Algorithmus die Parameter zwischen den ausgewählten Materialflusstrecken verbessert. Weitere Versuche müssen zeigen, inwiefern der Prozess der Ermittlung dieser "optimalen" Parameter durch Erweiterungen des eigentlichen Algorithmus beschleunigt werden kann. Darüber hinaus muss das Konzept in weiteren Studien mit deutlich komplexeren Modellen validiert werden.

Literatur

- d3FACT10: Projektgruppe d FACT: d FACT2.1.x Projektwiki.
<https://macabeo.cs.upb.de/projects/d3fact/wiki>, Stand: 11.08.2010.
- DANGELMAIER, W.; LAROQUE, C.: Immersive 3D-Ablaufsimulation von richtungsoffenen Materialflussmodellen zur integrierten Planung und Absicherung von Fertigungssystemen. In: Management komplexer Materialflüsse mittels Simulation. Hrsg.: ENGELHARDT-NOWITZKI, Corinna; NOWITZKI, Olaf; KRENN, Barbara. Wiesbaden: DUV Verlag, 2007.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A.: Ant System: An outocatalytic optimizing process. Working paper No. 91-016 Revised. Mailand: Politecnico di Milano, 1991.
- DORIGO, M.; STÜTZLE, A.: Ant Colony Optimization, MIT Press, 2004.
- GOSS, S.; ARON, S.; DENEUBOURG, J.-L.; PASTEELS, J. M.: Self-organized shortcuts in the Argentine ant. In: Berlin, Naturwissenschaften, 76(1989), S. 579-581.
- LAROQUE, C.: Ein mehrbenutzerfähiges Werkzeug zur Modellierung und richtungsoffenen Simulation von wahlweise funktions- und objektorientiert gegliederten Fertigungssystemen. Paderborn: HNI Verlagsschriftenreihe, 2007.
- LAW, A. M., KELTON W. D.: Simulation Modeling and Analysis. Bosten, MA: McGraw-Hill, 2000.
- VDI 4499 Blatt 1: Die Digitale Fabrik, Grundlagen. Februar 2008.