

Swarm Intelligence

Kai Bornemann

Otto-von-Guericke-Universität Magdeburg
kai.bornemann@st.ovgu.de

Zusammenfassung. Swarm Intelligence als Teilgebiet der künstlichen Intelligenz in der Informatik ist ein Lösungsansatz um Probleme zu lösen. Auf biologischer Grundlage von kollektivem Verhalten von sozialen Tieren inspiriert, werden dezentralisierte Agenten modelliert, die anhand von lokalen Informationen Entscheidungen treffen und als Kollektiv komplexe Aufgaben lösen. Vor allem bei diskreten und kontinuierlichen Optimierungsprobleme kommt Swarm Intelligence erfolgreich zum Einsatz. Dieses Paper gibt einen einführenden Einblick in die Swarm Intelligence, häufigen Einsatzgebieten und ihren biologischen Grundlagen. Anschließend werden bekannte Algorithmen der Swarm Intelligence beschrieben und ein tieferer Einblick in Ant Colony Optimization gegeben. Anschließend werden aktuelle Trends beschrieben.

1 Einführung

Swarm Intelligence (SI) ist ein Teilgebiet der künstlichen Intelligenz (AI). Die Grundidee bei SI ist die Adaption vom Verhalten von Individuen eines sozialen Kollektiv von Tieren. 1988 wurde SI das erste mal von Beni bei zellulärer Robotik verwendet. Agenten organisierten sich selbst durch die Interaktion mit ihren lokalen Nachbarn [2].

Seit dem hat sich SI stark weiterentwickelt und ist heute ein wichtiger Bestandteil in der Informatik.

In diesem Kapitel werden wichtige Anwendungen von SI in der Informatik beschrieben. Anschließend werden einige biologische Grundlagen, auf denen SI basiert, erläutert.

1.1 Anwendungen

Da es viele verschiedene Arten von kollektiven Tieren in der Natur gibt und viele verschiedene Spezies unterschiedliche Mechanismen verwenden (siehe Abschnitt 1.2), lassen sich SI Ansätze auf viele unterschiedliche Arten von Problemen in der Information anwenden. Nachfolgend gibt es eine Auswahl an Anwendungsgebieten [4].

Im allgemeinen wird SI vor allem in der stetigen, kontinuierlichen, dynamischen und stochastischen Optimierung genutzt.

SI finde Verwendung in der Planung von Fahrzeugrouten, bei der Lösung Graphenprobleme wie beispielsweise die Graphenfärbung, Traveling Salesman Problem und Cliquenbildung.

Weitere Anwendungsgebiete ist die Anordnung von Zellen in Schaltkreisen und im Design von Kommunikationsnetzwerken. Außerdem wird SI in der Molekularbiologie genutzt, um Proteinfaltungen zu finden. In der Robotik wird SI zur Orientierung und Pfadfindung von Robotern und zur Koordinierung von Multi-Robotersystemen genutzt.
TODO: weitere?

1.2 Biologische Grundlagen

Swarm Intelligence basiert auf Beobachtungen und Forschungen vom Verhalten von sozialen, kollektiven Tieren. Die Individuen einer solche Gruppe verhalten sich recht simpel, erledigen einfachen Aufgaben und lösen in der Gruppe komplexe Aufgaben.

Ameisen finden auf diese Art und Weise einen optimalen Weg zu einer Futterquelle und bauen komplexe Ameisenhögel. Bienen sind in der Lage große Gruppen zu einer neuen Futterquelle zu führen und sie errichten Bienenstöcke mit Waben, die zur Lagerung von Nahrung und zur Brut von Larven dienen, die nach bestimmten Regeln organisiert werden. Vögel-, Fisch- und Heuschreckenschwärme scheinen sich als ein einziger Organismus fortzubewegen, ohne miteinander zu kollidieren und überwinden so weite Entfernungen zu einem gemeinsamen Zielort [1,4,7]. Die zugrunde liegenden Mechanismen für das Schwarmverhalten der verschiedenen Spezies sind hierbei unterschiedlich. Im Folgenden werden einige allgemeine Mechanismen und Beispiele für deren Anwendung erläutert.

Direkte Kommunikation: Damit eine soziales Kollektiv von Tieren eine Futterquelle erschließen kann, muss ein optimaler Weg zu einer ausreichend hochwertigen Futterquelle erschlossen werden. Da es keine zentrale Regierung in solchen Gruppen gibt, müssen die Individuen sich auf Informationen ihrer Umgebung und ihren direkten Nachbarn verlassen und diese sinnvoll nutzen.

Die Kommunikation von Ameisen zur Erschließung von Futterquellen findet auf indirektem Wege statt.

Sie sondern jederzeit Pheromone ab, die anderen Ameisen zur Orientierung dienen. Die so entstehenden Pheromone sind je nach Befinden der Ameise unterschiedlich stark. Wenn ein Individuum eine qualitativ hochwertige Futterquelle findet, steigt die Stärke der Pheromone an, wenn sie keine Futterquelle findet, bleibt sie gering.

Wenn bisher keine Futterquelle erschlossen wurde, laufen die Ameisen zufällig in alle Richtungen, um eine solche Quelle zu finden. Bei Erfolg fressen sie davon und nehmen eine Probe davon mit zu ihrem Bau, hierbei hinterlassen sie bereits eine erhöhte Pheromonspur. Andere Ameisen wittern diese Spur und die Wahrscheinlichkeit erhöht sich, dass sie diese ebenfalls nutzen. Bei Nutzung dieses Pfades verstärken sie den Pfad weiterhin.

Schlechtere Pfade verlieren immer weiter an Pheromonen, da nur wenige Ameisen diesen Weg nutzen und die Wahrscheinlichkeit immer geringer wird, dass weitere Ameisen der Pheromonspur folgen. Nach einiger Zeit hat sich ein Pfad zu einer Futterquelle etabliert, den die Kolonie nutzt.

Dieser indirekte Prozess der Kommunikation ermöglicht es Ameisen den optimalen Pfad zu einer möglichst hochwertigen Futterquelle zu finden [4].

Indirekte Kommunikation: Andere Spezies wie beispielsweise Honigbienen nutzen eine direkte Art der Kommunikation zur Erschließung von Futterquellen.

Ähnlich wie bei den Ameisen fliegen die Bienen initial zufällig durch die Gegend und suchen aktiv nach Futterquellen. Bienen, die diese Aufgabe verrichten werden Scouts genannt. Wenn ein Scout eine Futterquelle findet, fliegt er zurück zu seinem Nest und informiert andere Bienen über die Futterquelle. Hierbei führt der Scout einen Tanz auf, der die anderen Bienen über die Richtung, die Entfernung und die Qualität der Futterquelle informiert. Wenn andere Bienen von der Futterquelle überzeugt wurden, tanzen sie ebenfalls für diese. Nach einiger Zeit konvergieren die Tänze der Bienen auf eine einheitliche Futterquelle, die Anzahl der Scouts nimmt ab und die Anzahl der Bienen, die das Futter für den Stock sammeln, nimmt zu [4].

Exploration und Exploitation: Eine Futterquelle für eine Spezies ist natürlicherweise beschränkt, das heißt sie kann versiegen oder durch Ereignisse nicht mehr zur Verfügung stehen. Es besteht ebenfalls die Möglichkeit, dass eine neue, bessere Futterquelle entsteht. In solchen Fällen ist es unerlässlich für eine Spezies in der Lage zu sein eine neue Futterquelle zu erschließen.

Das bedeutet, dass ein Individuum einer Gruppe in der Lage sein muss bestehende Informationen zu speichern und nutzen (Exploitation), gleichzeitig aber auch die Möglichkeit besitzen muss neue Informationen zu erschließen (Exploration). Ein Individuum muss also einen Kompromiss dieser beiden Konzepte eingehen.

Wenn ein Ameisenstamm eine etablierte Futterquelle, durch die etablierten Pheromonspur, nutzt (Exploitation), besteht jederzeit die Chance, dass eine Ameise sich gegen diese Spur entscheidet und wieder zu einem Scout wird, der eine andere Futterquelle finden kann (Exploration). Bei Erfolg erhöht dies die Chance, dass weitere Ameisen dieser Spur folgen, da die Pheromonspur für diese alternative Quelle immer stärker wird, sofern die Futterquelle von ausreichender Qualität ist. Auf diese Art und Weise besteht jederzeit die Möglichkeit eine alte Futterquelle zu verlassen und eine neue, hochwertigere zu erschließen [4].

Gruppenbewegungen: Damit sich große Tierschwärme, wie Fische, Vögel und Heuschrecken ohne Kollisionen zielgerichtet fortbewegen können, halten sich die Individuen solcher Schwärme bestimmte Regeln ein [1].

1. **Homogenität:** Jedes Individuum hält sich an dieselben Regeln. Der Schwarm hat meist keinen Führer, temporäre Führer sind möglich. Ein Führer ist ein Gruppenmitglied, das Informationen über das Ziel besitzt.

2. **Lokalität:** Die Bewegung eines Individuums wird nur durch die Bewegungen von direkten Nachbarn beeinflusst.
3. **Kollision:** Jedes Individuum vermeidet Kollision mit seinen unmittelbaren Nachbarn.
4. **Geschwindigkeitsangleichung:** Jedes Individuum gleicht seine Geschwindigkeit an die Geschwindigkeiten seiner Nachbarn an.
5. **Schwarmzentrierung:** Sie versuchen eine möglichst geringe Distanz zu ihren Nachbarn zu halten, ohne andere Regeln zu verletzen.

Hierbei stellt sich weiterhin die Fragen, wie eine solche Gruppe sich über ein gemeinsames Ziel einigt. Hierbei gibt es zwei Möglichkeiten:

1. Eine solche Gruppe besitzt informierte Führer (beispielsweise nur um die 5% bei Bienen), die die uninformierten Individuen leiten.
2. Jedes Gruppenmitglied trägt zur Richtung des Schwarms bei, dies setzt voraus, dass jedes Individuum ungefähr dasselbe Ziel besitzt.

Bei geführten Gruppen, müssen die führenden Individuen die uninformierten Individuen über das gemeinsame Ziel informieren. Dies kann entweder implizit oder explizit geschehen:

1. **Explizit:** Die Führer machen sich explizit bemerkbar, indem sie sich schneller in Richtung des Ziels bewegen als die uninformierten Individuen, wodurch diese Mitglieder einen höheren Einfluss auf die Gruppenbewegung des Schwarmes haben, als uninformierte.
2. **Implizit:** Die Führer lenken den Schwarm implizit, da sie die einzigen sind, die wissen, wo sich das Ziel befindet. Die uninformierten Individuen gleichen sich unwissentlich an die Führer an. Keinem der Schwarmmitglieder ist klar, dass es führende Individuen gibt.

2 Algorithmen

In diesem Kapitel werden Algorithmen in der SI diskutiert.

Zwei der bekanntesten SI Algorithmen sind Ant Colony Optimization (ACO) und Particle Swarm Optimization (PSO) [1]. Dieses Kapitel beginnt mit einer genaueren Erläuterung von Ant Colony Optimization und einer Diskussion zu dessen Anwendungsgebieten, Schwächen und Stärken. Anschließend wird eine Übersicht über weitere SI Algorithmen und deren grundlegende Idee vermittelt.

2.1 Ant Colony Optimization

Ant Colony Optimization (ACO) wurde in den frühen 90ern von Marco Dorigo und seinen Kollegen zur Lösung von diskreten Kombinationsproblemen eingeführt und war der erste SI Algorithmus, der als Annäherungsansatz von Optimierungsproblemen eingesetzt wurde [11,4].

ACO basiert im Kern auf der indirekten Kommunikation zwischen Ameisen durch zurückgelassene Pheromone, durch die sie den optimalen Weg

zu einer Futterquelle finden können (siehe Abschnitt 1.2).
 Im Folgenden wird ein allgemeines ACO Framework beschrieben, um kombinatorischen Optimierungsprobleme (CO) zu lösen [9].

Das ACO Framework: Bei der ACO sind im allgemeinen folgende Schritte zum Finden von potenziellen Lösungen wichtig [4]:

1. **Lösungskomponenten:** Bilden einer Menge $C = \{c_1, \dots, c_n\}$ von Lösungskomponenten c_i , um Lösungen für das Kombinationsproblem zu finden. Diese werden von dem konkreten Problem, das es zu lösen gilt, abgeleitet.
2. **Pheromonmodell:** Definiere eine Menge von Pheromonwerten $T = \{t_1, \dots, t_n\}$. Jeder Pheromonwert t_i wird mit einer Lösungskomponente c_i assoziiert.
3. **Lösungssequenz:** Jede Konstruktion einer möglichen Lösung beginnt mit einer leeren Lösungssequenz $s = \langle \rangle$. Diese wird in jedem Konstruktionsschritt um eine Lösungskomponente c_i erweitert. Alle verbleibenden möglichen Lösungskomponenten werden durch die Menge $N(s) \subseteq C \setminus s$ beschrieben. Die genaue Auswahl der verbleibenden Lösungskandidaten sind Problem- und Anwendungsabhängig.
4. **Auswahl der Lösungskomponenten:** Die Lösungskomponenten werden anhand ihrer assoziierten Pheromonwerte gewählt. Die Wahrscheinlichkeit $p(c_i)$, dass eine Lösungskomponente c_i gewählt wird, wird im Allgemeinen mit der folgenden Übergangswahrscheinlichkeit ausgedrückt:

$$p(c_i) = \frac{(t_i)^\alpha \cdot w(c_i)^\beta}{\sum_{c_j \in N(s)} (t_j)^\alpha \cdot w(c_j)^\beta}, \forall c_i \in N(s) \quad (1)$$

$w(c_i)$ ist eine Gewichtungsfunktion, die bestimmte Lösungskomponenten stärker oder schwächer gewichten kann. Die Exponenten α und β sind positive Werte, die angeben wie stark der Gewichtung und des Pheromonwertes vertraut werden soll. Die Division durch die Summe aller restlichen Lösungskandidaten $N(s)$ normalisiert die Summe aller Wahrscheinlichkeiten $p(c_i)$ auf eins.

5. **Aktualisieren der Pheromonwerte:** Nachdem eine neue Lösungskomponente in die Lösungssequenz aufgenommen wurde, werden alle relevanten Pheromonwerte aktualisiert. Die Aktualisierung besteht aus zwei Teilen. Erstens, einer Evaporation, die alle Pheromonwerte mit jedem Iterationsschritt verblassen lässt. Zweitens, die Erhöhung der Pheromonwerte, die Teil der aktuellen Lösungssequenz sind.

$$t_i \leftarrow (1 - p) \cdot t_i + p \cdot \sum_{s \in S_u} w_s \cdot F(s) \quad (2)$$

p ist hierbei die Evaporationsrate aus dem Intervall $(0, 1]$, die für alle t_i konstant ist. Dies verhindert eine vorzeitige Konvergenz auf

eine suboptimale Lösung. Sie kann auch als Teil der Exploration von weiteren Lösungskandidaten angesehen werden.

$F(s)$ ist eine Qualitätsfunktion, die eine Lösungssequenz bewertet. w_s ist ein positives Gewicht für die Qualität einer Lösungssequenz. S_u sind alle Lösungssequenzen, die für die Aktualisierung relevant sind.

6. **Iteration:** Iteriere die Schritte 4. und 5., bis der Algorithmus konvergiert oder ein Abbruchkriterium erfüllt wurde.

Um ACO auf ein bestimmtes Problem anzuwenden, sind vor allem Veränderungen in der Qualitätsfunktion $F(s)$, den verschiedenen Gewichten α , β , w_s und $w(c_i)$ und der Definition von relevanten Teillösungen S_u nötig.

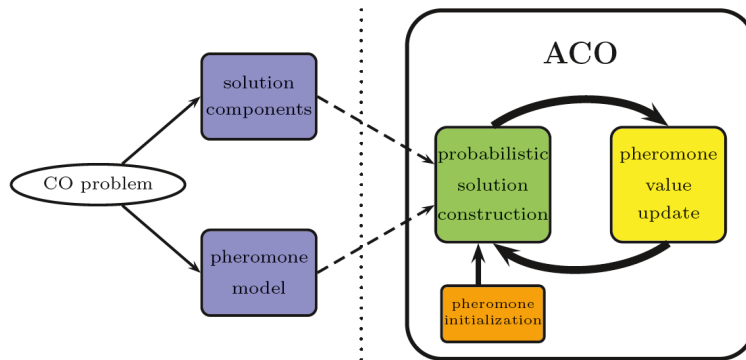


Abb. 1. Schematische Übersicht zum ACO Framework nach Beni [2,4].

Diskussion: ACO wurde erfolgreich zur Lösung von vielen klassischen Problemen wie beispielsweise der Planung von Fahrzeugrouten, Graphenfärbung, Zeitplanoptimierung und maximale Clique Problem angewandt.

Anfänglich schnitt ACO jedoch schlechter ab als die state-of-the-art Algorithmen. In den folgenden Jahren wurde viele Erweiterungen und Veränderungen am Algorithmus vorgenommen. Diese Verbesserungen beinhalteten fast ausschließlich Veränderungen der Regeln für die Aktualisierung der Pheromone. Elitist AS (EAS) [11], Rank-based AS (RAS) [8], MAX-MIN Ant System (MMAS) [16], Ant Colony Systems (ACS) [10] und Hyper-CUbe Framework (HCF) [3] sind einige dieser verbesserten Versionen von ACO.

Aktuelle Anwendungen von ACO beinhaltet unter anderem: Platzierung von Zellen in Schaltkreisen, Design von Kommunikationsnetzwerken, Probleme in stetiger Optimierung, dynamische und stochastische

Probleme.

ACO ist mittlerweile ein state-of-the-art Algorithmus um sequentielle Sortierungsprobleme und Ressourcen Verteilungsprobleme zu lösen und 2D und 3D hydrophobische polare Proteinfaltungen zu finden [4,15].

Zudem existieren viele hybride Algorithmen, die ACO in andere Ansätze integrieren. Beispiele sind ACO mit Branch & Bound Derivatives [5], ACO mit Constraint Programming [14] und ACO in Multilevel Frameworks [13,6].

2.2 Weitere Algorithmen

ACO ist nicht der einzige SI Algorithmus der durch das Verhalten von sozialen Tiergruppen inspiriert wurde. Nachfolgend befindet sich eine kleine Auswahl von SI Algorithmen und deren grundlegender Ansatz.

1. **Particle Swarm Optimization:** Ein weiterer populärer SI Algorithmus ist Particle Swarm Optimization (PSO). Dieser Algorithmus orientiert sich an der Gruppenbewegung (siehe Abschnitt 1.2) von Vögel- und Fischeschwärmen. Erstmals wurde PSO 1995 von Eberhart beschrieben [12]. PSO ist ein robuster und effizienter Algorithmus zur Lösung von Optimierungsproblemen.

Bei diesem Algorithmus werden Partikel eines Schwarms in einem Suchraum ausgesetzt, in dem sie sich mit einer bestimmten Geschwindigkeit bewegen können. Jede Position des Partikels stellt dabei eine mögliche Lösung des Problems dar. Das Ziel ist es nun den Suchraum nach einer optimalen oder guten Lösung abzusuchen. Dazu interagieren die Partikel mit ihren Nachbarn, teilen und nutzen ihre eigenen Informationen und verändern ihre Position anhand von diesen Informationen. Nach und nach konvergieren alle Partikel zu einem gemeinsamen Ort hoher Qualität, von dem die Partikel annehmen, dass er das Optimum darstellt.

PSO ist vor allem sehr gut darin dynamische Optimierungsprobleme zu lösen. Dies sind Probleme, bei denen sich die Qualität einer Lösung über die Zeit verändert [4].

2. **Artificial Bee Colony Algorithm:** TODO
3. **Artificial Swarm Intelligence:** TODO
4. **Glowworm Swarm Optimization:** TODO

3 Auswertung und Ausblick

TODO

Literatur

1. Abraham, A., Grosan, C., Ramos, V.: Swarm Intelligence in Data Mining. Studies in Computational Intelligence, Springer Berlin Heidelberg (2007)
2. Beni, G.: The concept of cellular robotic system. In: Intelligent Control, 1988. Proceedings., IEEE International Symposium on. pp. 57–62 (Aug 1988)
3. Blum, C., Dorigo, M.: The hyper-cube framework for ant colony optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34(2), 1161–1172 (April 2004)
4. Blum, C., Merkle, D.: Swarm Intelligence: Introduction and Applications. Natural Computing Series, Springer Berlin Heidelberg (2008)
5. Blum, C.: Beam-aco: Hybridizing ant colony optimization with beam search: An application to open shop scheduling. Comput. Oper. Res. 32(6), 1565–1591 (Jun 2005)
6. Blum, C., Vallès, M.Y.: Multi-level Ant Colony Optimization for DNA Sequencing by Hybridization, pp. 94–109. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Proceedings volume in the Santa Fe Institute studies in the sciences of complexity, OUP USA (1999)
8. Bullnheimer, B., Hartl, R.F., Strauss, C.: A new rank based version of the ant system. a computational study. Central European Journal of Operations Research 7(1), 25–38 (1999)
9. Dorigo, M., Caro, G.D.: Ant colony optimization: a new meta-heuristic. In: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. vol. 2, p. 1477 Vol. 2 (1999)
10. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (Apr 1997)
11. Dorigo, M., Maniezzo, V., Coloni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26(1), 29–41 (Feb 1996)
12. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: In Proceedings of the 1995 IEEE International Conference on Neural Networks. vol. 4, p. 1942–1948 (1995)
13. Korošec, P., Šilc, J., Robič, B.: Mesh-Partitioning with the Multiple Ant-Colony Algorithm, pp. 430–431. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
14. Meyer, B., Ernst, A.: Integrating ACO and Constraint Propagation, pp. 166–177. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
15. Shmygelska, A., Hoos, H.H.: An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. BMC Bioinformatics 6(1), 1–22 (2005)
16. Stützle, T., Hoos, H.H.: Max–min ant system. Future generation computer systems 16(8), 889–914 (2000)