

Schwarmintelligenz

Lydia Pintscher

Universität Karlsruhe

Zusammenfassung Dieses Paper soll die Grundgedanken der Schwarmintelligenz erläutern und wie sie genutzt werden kann um Probleme in einer immer komplexer werdenden Welt zu lösen. Dabei wird sowohl auf Beispiele aus der Natur als auch deren Anwendung in der Informatik eingegangen.

„Was nicht gut für den Schwarm ist, ist nicht gut für die Biene.“
- Markus Aurelius

1 Grundlagen der Schwarmintelligenz

Die Natur verblüfft uns immer wieder mit hochkomplexem Verhalten bei Tieren in großen Gruppen. Angefangen bei der blitzartigen Bewegung eines ganzen Fischschwarms, über die Nahrungssuche der Ameisen und Bienen bis hin zum Fluchtverhalten einer Elchgruppe. Bei genauerem Hinschauen beruht dieses Verhalten aber zumeist auf wenigen sehr simplen Regeln.

Doch nicht nur dort findet man intelligente Schwärme. Sie findet sich auch im Verhalten des Menschen wieder und wird als neuer Lösungsansatz für Probleme genutzt, die mit herkömmlichen Algorithmen nur unzureichend lösbar sind.

Schwarmintelligenz lässt sich daher charakterisieren als Verhalten von vielen Individuen, die nach einfachen Regeln handeln. Als Ergebnis dessen entstehen komplexe Verhaltensmuster, die allein auf Grundlage der einfachen zugrunde liegenden Regeln nicht immer erkennbar sind und zu denen die einzelnen Individuen nicht fähig wären. Diese Gruppen zeichnen sich durch Selbstorganisation¹, Anpassungsfähigkeit und Robustheit² aus.

Intelligenz zeigt sich hier nicht, wie im üblichen Intelligenzbegriff beim Menschen, als Fähigkeit Fakten und Zusammenhänge zu verstehen, neues Wissen zu erwerben und dieses zur Lösung von Problemen zu nutzen. Vielmehr beschreibt Intelligenz hier die Fähigkeit ein Gruppengedächtnis aufzubauen und Probleme in einer Gruppe zu lösen, indem jedes Mitglied einen Teil dazu beiträgt. Im Gegensatz zum Menschen ist hier die Zusammenarbeit in der Gruppe zum Erreichen des Ziels von essentieller Bedeutung.

¹ nicht überwachte und dezentrale Koordination von Aufgaben

² Erfüllen der Aufgaben trotz Ausfall einiger Individuen

2 Nutzung der Schwarmintelligenz

2.1 Schwarmintelligenz in der Natur

Das Konzept der Schwarmintelligenz hat seine Vorbilder in der Natur. Zum Verständnis seiner späteren Anwendung werden in diesem und dem folgenden Abschnitt einige natürliche Ausprägungen von Schwarmintelligenz vorgestellt.

Ameisen sind ein Paradebeispiel für Schwarmintelligenz und eines der Vorbilder für deren Nutzung in der Informatik. Eine einzelne Ameise ist nicht besonders intelligent und kann keine komplizierten Aufgaben bewältigen. Im Gegensatz dazu vollbringt ein ganzer Ameisenstaat Erstaunliches und passt sich gut an neue Gegebenheiten seiner Umwelt an. Sie bilden Staaten mit einigen hundert bis zu mehreren Millionen Individuen. Trotz dieser riesigen Anzahl funktioniert ein Ameisenstaat erstaunlich gut, da er sich selbst organisiert ohne ein Individuum, das einen Überblick über alle Aufgaben hat oder diese sogar verteilt. Stattdessen führen die Handlungen einzelner Ameisen im Zusammenspiel zu einem organisierten Staat, der für sie sorgt und Nahrung, Brutpflege, Schutz und mehr zur Verfügung stellt. Für die Informatik besonders interessant sind sie allerdings aufgrund ihrer Fähigkeiten effiziente Wege zu Nahrungsquellen zu suchen. Einzelne Ameisen gehen auf die Suche nach neuen Nahrungsquellen und hinterlassen dabei auf dem Hinweg zur Quelle und dem Rückweg zum Ameisenbau Pheromone, die mit der Zeit verfliegen. Je öfter eine oder mehrere Ameisen innerhalb eines kurzen Zeitraums auf dem selben Weg gehen, umso intensiver ist diese Markierung. Dies führt dazu, dass immer mehr Ameisen diesen Weg benutzen, dabei erneut Pheromone hinterlassen und weitere Nahrung in den Bau transportieren bis die Quelle erschöpft ist. Ein kurzer Weg zu einer guten Nahrungsquelle wird also bevorzugt, da die Pheromonspur hier besonders intensiv ist. Für einen großen Ameisenstaat ist das eine sehr effiziente Methode, die ohne Intervention und Management einer einzelnen höheren Instanz auskommt und nur auf lokalen Informationen basiert. Eine einzelne Ameise muss keinen Überblick über alle Vorgänge im Staat haben um zu entscheiden an welchem Ort sie nach Nahrung suchen soll. Sie trifft diese Entscheidung allein auf Grundlage der Pheromonspur und dadurch mittels Kommunikation mit anderen Ameisen.

Bienen nutzen Schwarmintelligenz bei der Nahrungssuche. Kundschafterinnen und Sammlerinnen, die von der Pollensuche zurück kommen, führen einen Tanz auf und zeigen damit Richtung und Entfernung der Nahrungsquelle mit der Richtung und Geschwindigkeit ihres Tanzes in Abhängigkeit vom Sonnenstand an [1], wie in Bild 1 dargestellt. Die Anzahl der Wiederholungen zeigt dabei die Entfernung und der Winkel die Richtung in Bezug zur Sonne an. Unbeschäftigte Bienen sehen sich die Tänze ankommender Bienen an und wählen ihr nächstes Ziel. Sobald eine Blüte nicht mehr sehr ergiebig ist brauchen die Sammlerinnen länger beim Sammeln und veranlassen aufgrund ihres weniger häufigen Tanzens weniger weitere Sammlerinnen mit ihnen am gleichen Ort Nahrung zu suchen. Die Bienen treffen diese Entscheidungen wieder nur aufgrund lokaler Informationen und sind nicht von einer Instanz, die Arbeit zuteilt, abhängig. Bei der Suche nach einem neuen Nest vertrauen Bienen ebenfalls auf den Schwarm. Viele Bie-

nen fliegen aus und suchen einen guten neuen Platz. Sobald eine Stelle etwa 15 Bienen angezogen hat kommen die anderen dazu und sie informieren den Rest des Schwarms darüber. Der Schwarm hat sich entschieden. Diese Strategie ist sehr erfolgreich, da zuerst viele Optionen völlig unbeeinflusst von den anderen Bienen abgewägt werden, eine einfach Strategie genutzt wird um die Anzahl der Entscheidungsmöglichkeiten zu verringern und dann eine Mehrheit entscheidet.

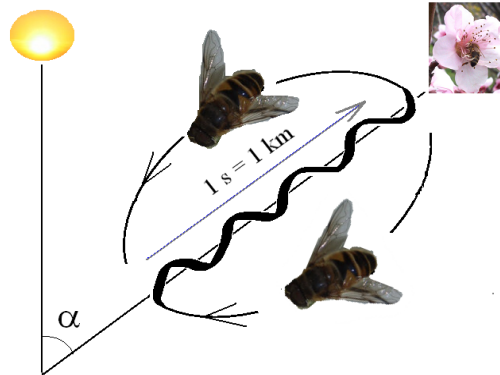


Abbildung 1. Der Bientanz [2]

Schwarmfische sind ein erstaunliches Beispiel für schnelle Reaktion auf äußere Einflüsse. Ein Leben im Schwarm hat verschiedene Vorteile für den einzelnen Fisch. Ein Angreifer wird wahrscheinlich schneller entdeckt von einer großen Gruppe als von einem Individuum; bei einem Angriff wird der Angreifer verwirrt, da es schwerer ist einen einzelnen Fisch im Schwarm auszumachen und zu jagen und zuletzt lassen sich Angreifer täuschen und glauben der Schwarm sei ein einziger größerer Fisch. Um eben diese Vorteile des Schwarms ausnutzen zu können ist es von essenzieller Bedeutung, dass sich der Schwarm einheitlich bewegt. In [3] wird gezeigt, dass sich Fische im Schwarm an ihren Nachbarn orientieren und zu ihnen einen bestimmten Abstand halten um Kollisionen beziehungsweise ein Auseinanderbrechen des Schwarms zu vermeiden. Dabei hilft ihnen ihr Seitenlinienorgan. Ebenso passen sie ihre Geschwindigkeit an die der benachbarten Fische an. Sie folgen also den einfachen Regeln „Halte dich in einem bestimmten Abstand zu anderen Fischen auf“ und „Schwimme so schnell wie deine Nachbarn“. Reynoldz fasst dies zusammen als Separation³, Alignment⁴ und Cohesion⁵ [4]. Im Angriffsfall entsteht hieraus abhängig von der Art sehr komplexes Verhalten. Einige Fischarten lassen den Schwarm „explodieren“, indem alle Fische geordnet auseinanderströmen; andere schlagen einen Haken und begeben sich hinter den Angreifer; wieder andere teilen den Schwarm und zwingen den Angreifer somit

³ Individuum entfernt sich wenn andere Individuen zu nahe sind

⁴ Individuum begibt sich in die Durchschnittsrichtung seiner Nachbarn

⁵ Individuum versucht die Durchschnittsposition seiner Nachbarn zu erreichen

sich für einen Teil zu entscheiden. Auch hier gibt es wieder keine Instanz, die den Schwarm lenkt. Vielmehr beeinflussen die Fische, die von der Gefahr wissen implizit durch ihr eigenes Fluchtverhalten den ganzen Schwarm.

Ein ähnliches Verhalten ist bei Elchen zu beobachten. Sie wandern in Gruppen. Befindet sich ein Wolf in nicht unmittelbarer Nähe sind sie aufmerksam. Kommt er über eine bestimmte Schwelle sind sie in Alarmbereitschaft versetzt und bereiten sich auf seinen Angriff vor. Doch erst wenn er eine weitere Schwelle überschreitet ergreift ein Tier nach dem anderen geordnet, wie in einer Welle, die durch die Herde geht, die Flucht. Diese geordnete Flucht verwirrt den Angreifer und kommt jedem einzelnen Elch zu gute da kein Chaos entsteht. Jeder Elch weiß wann und in welche Richtung er fliehen muss um die Herde nicht zu gefährden und selbst zu entkommen.

Die Untersuchung des Schwarmverhaltens legt natürlich eine Manipulation dessen nahe. Schaben sind Gruppentiere und halten sich gern zusammen in dunklen Bereichen auf. Halloy et al haben in [5] eine künstliche Schabe in eine Gruppe Schaben gebracht. Sie stellten zwei verdunkelte Plätze zur Verfügung und die Schaben zogen sich gemeinsam zurück in einen davon. Man geht davon aus, dass sie nach der einfachen Regel „Laufe umher und verbringe mehr Zeit an dunklen Orten an denen es andere Schaben gibt“ handeln um einen Platz zu wählen. Die künstlichen Schaben wurden aufgrund eines ähnlichen Dufts angenommen, interagierten mit den echten und fanden einen Ruheplatz. Nun wurden sie so programmiert, dass sie nicht mehr dunkle Plätze aufsuchen sondern helle. Nach nicht langer Zeit befanden sich auch die echten Schaben im hellen Unterschlupf. Im Experiment mit 12 echten Schaben waren dazu vier Roboterschaben nötig. Dieses Experiment zeigt, dass es möglich ist Schwarmverhalten aktiv zu beeinflussen. Vielleicht ist es ja schon bald nicht mehr nötig Schaben mit wenig Erfolg versprechendem Gift zu behandeln, sondern man bringt sie völlig biologisch unbedenklich dazu sich an einen Ort zu versammeln an dem man sie einsammeln und beseitigen kann.

2.2 Schwarmintelligenz beim Menschen

Auch der Mensch macht sich Schwarmintelligenz zu Nutze und zeigt sie.

Auf der Internetseite digg.com geben täglich tausende Nutzer Links zu Artikeln, Videos und Podcasts ihr Vote (genannt digg) die zuvor von anderen eingepflegt wurden. Umso mehr diggs ein Link bekommt umso populärer ist er und umso weiter vorn erscheint er auf digg.com. Mit diesem einfachen System ist es den Gründern von Digg gelungen die Art wie ihre Nutzer das Internet sehen zu verändern. Sie erhalten Zugang zu Informationen auf die sie sonst mit großer Wahrscheinlichkeit nicht gestoßen wären. Aufgrund der großen Nutzerbasis wird ein breites Spektrum an Informationsquellen abgedeckt und so einer großen Masse zugänglich gemacht und für sie gefiltert. Doch es gibt auch Kritikpunkte an Digg. Immer wieder werden Stimmen laut, die sagen, dass Digg nicht unvoreingenommen ist und die Nutzer sich ihre eigene Welt diggen. Dies ist sicher berechtigt und liegt darin begründet, dass die Zielgruppe junge technikaffine Männer sind, sogenannte early adopters. Mit einem zunehmenden Ausbrechen

aus diesem starren Gefüge in den letzten Monaten sind aber verstärkt zum Beispiel konservative Artikel mit vielen diggs zu beobachten, die nicht dem üblichen Schema dessen entsprechen was Digg-Nutzer bevorzugen. Der zweite Kritikpunkt der immer wieder genannt wird, sind die Kommentare die Nutzer zu jedem Link abgeben können. Digg zieht vermehrt sogenannte Trolle⁶ an. Hiergegen setzt man wieder auf ein Bewertungssystem. Nutzer können hierbei Kommentare positiv oder negativ bewerten und Kommentare ab einem bestimmten Schwellwert nicht mehr anzeigen lassen. Dies hat zur Folge, dass Trolle nicht mehr die gesuchte Aufmerksamkeit der breiten Masse erhalten. Wirklich beseitigen kann man das Problem auf diese Weise aber nicht, wie eindrucksvoll tagtäglich demonstriert wird. Das Hauptproblem ist hierbei wohl das Verständniss des Bewertungssystems durch die Community. Nutzer bewerten oft nach Übereinstimmung mit den eigenen Meinungen und nicht darauf basierend ob ein Kommentar objektiv gesehen „gut“ oder „schlecht“ ist. Ein Beispiel gelungener Kommentarmoderation durch eine breite Nutzerbasis ist Slashdot. Hier darf allerdings nur moderieren wer positives Karma hat, das heißt sich verdient gemacht hat durch aufschlussreiche Kommentare zum Beispiel. Durch diese kleine Einschränkung hat es Slashdot geschafft bei einer ähnlichen (wenn auch nicht identischen) Zielgruppe wesentlich bessere Diskussionen in den Kommentaren zu fördern und bewegt sich von einer Bewertung nach Übereinstimmung mit der Weltanschauung des Moderators weg.

Google macht sich die Intelligenz der Masse zu Nutze. Bei der Berechnung des Pagerank, der maßgeblich für die Platzierung einer Seite innerhalb der Suchergebnisse verantwortlich ist, bestimmt Google die Popularität unter anderem anhand der eingehenden Links auf diese Seite, welche andere Webmaster, Redakteure und Nutzer gesetzt haben. Google vertraut allerdings nicht nur darauf, da eine oft verlinkte Seite nicht automatisch ein gutes Suchergebniss bedeutet. Gründe hierfür sind Linkfarmen⁷ und minderwertiger Content.

J. Krause und J. Dyer wollten ihre Theorien zu Schwarmverhalten testen. Sie gingen davon aus, dass nur wenige einfache Regeln nötig sind um es entstehen zu lassen. Dazu haben sie im April 2007 zusammen mit dem Team von Quarks und Co ein Experiment mit 300 Freiwilligen durchgeführt [6]. Im ersten Versuch bekam jeder Teilnehmer die Anweisungen „Bleibe immer in Bewegung“ und „Bleibe immer etwa eine Armlänge entfernt von den anderen Schwärmen um dich herum“ und durfte nicht kommunizieren. Aufgrund dieser Regeln bildete sich schnell ein Torus aus zwei in entgegengesetzte Richtung laufenden Strömen. Im zweiten Versuch galten die Regeln des ersten und es wurde ein Räuber eingeführt. Von diesem mussten sich die Teilnehmer entfernen und mindestens zwei Armlängen Abstand halten. Vor dem Räuber, der immer die nächstgelegene „Beute“ jagte, teilte sich der Schwarm und schloss sich hinter ihm wieder, genau wie ein Fischwarm, der von einem Raubfisch angegriffen wird. Im dritten

⁶ Personen, die kontroverse Kommentare schreiben mit dem alleinigen Ziel Diskussionen zu erzwingen und Unruhe zu stiften

⁷ Massenverlinkung mit dem Ziel den Pagerank bei Google und anderen Suchmaschinen zu steigern; oft mit kommerziellen Zielen

Versuch soll herausgefunden werden wie viele Teilnehmer von 200 nötig sind um den Schwarm zu einer bestimmten Stelle zu bewegen. Hierzu galten wieder die Ausgangsregeln und fünf Teilnehmer bekamen verdeckt eine Stelle zugewiesen zu der der Schwarm zu leiten ist. Die fünf bewegten sich zu der Stelle, konnten aber den Rest nicht dazu bewegen zu folgen. Der Versuch wurde mit zehn wiederholt. Diesmal gelang es den Schwarm zu führen. Dies zeigt, dass schon zehn Prozent eines Schwarms ihn leiten können, wenn sie Informationen haben die ihnen einen Vorteil bringen⁸. Im letzten Versuch sollte getestet werden, was geschieht wenn im Schwarm zwei Gruppen mit unterschiedlichen Zielen versuchen zu leiten. Hierzu bekamen 20 Teilnehmer einen Ort genannt und 10 weitere einen anderen. Erwartet hatte man, dass sich der Schwarm dann für die Mehrheit entscheiden würde. Das passierte aber nicht, sondern der Schwarm bewegte sich zwischen den beiden Orten. In der Natur hat dieses Verhalten den Vorteil, dass noch mehr Individuen beide Alternativen sehen und sich entscheiden. Im Laufe der Zeit wird sich eine Mehrheit bilden. Mit diesen Versuchen wurde gezeigt, dass Schwarmverhalten tatsächlich aus den einfachen Regeln hervorgehen kann, die man vermutete.

2.3 Schwarmintelligenz in der Informatik und Technik

Schwarmintelligenz ist in der Natur, wie die vorherigen beiden Abschnitte erläutern, ein sehr erfolgreiches Konzept. In diesem Abschnitt soll auf die Übertragung und Anwendung dieser Prinzipien in der Informatik und Technik eingegangen werden.

Schwarmintelligenz findet in der Informatik bisher algorithmische Anwendung in Form von Ant Colony Optimization (ACO) und Particle Swarm Optimization (PSO) sowie im technischen Bereich in Form von Schwarmrobotern, auch Swarm-Bots genannt.

Schwärme sind hier besonders interessant, da sie aus sehr einfachen und leicht zu programmierenden beziehungsweise herzustellenden Teilen bestehen und äußerst komplexe Aufgaben bewältigen können. Macht einer dieser Teile des Schwarms einen Fehler oder fällt gar ganz aus führt dies nicht zum Scheitern der ganzen Mission, zum Nichterfüllen der Aufgabe oder nicht Lösen des Problems.

Ant Colony Optimization (ACO) [7] ist ein Ansatz das Verhalten der Ameisen, wie es in Abschnitt 2.1 beschrieben wurde, zu nutzen um kurze Wege in Graphen zu finden. Ein Anwendungsfall ist das Traveling Salesman Problem⁹. Hierbei werden Agenten ausgesickt um den Graph zu erkunden. Sie starten an zufällig gewählten Punkten und suchen einen Weg. An jeder Kreuzung entscheiden sie sich abhängig von der Pheromonmenge, die andere Agenten zurückgelassen haben, zufällig für den weiteren Weg. Haben alle einen Weg gefunden werden zuerst die Pheromonwerte aller Wege vermindert¹⁰ und dann entlang der gefundenen

⁸ in diesem Fall ein genaues Ziel im Gegensatz zu ziellosem Umherlaufen

⁹ Finden des kürzesten Weges in einem Graphen der alle Punkte genau einmal enthält und wieder zu zum Ausgangspunkt zurückführt

¹⁰ zur Simulation des Verfliegens der Pheromone in der Natur

Pfade entsprechend der Lösungsgüte angepasst. Dadurch entsteht ein globales Gedächtnis für eine gute Lösung. Diese Schritte werden so lange wiederholt bis eine bestimmte Abbruchbedingung/Lösungsgüte erreicht ist. Codeausschnitt 1 zeigt eine Pseudocodeimplementierung von Ant System (für das Traveling Salesman Problem) [8], der ersten Anwendung von Ant Colony Optimization. Die Wahrscheinlichkeit, dass eine Ameise von Stadt i nach Stadt j aus der Menge J der noch zu besuchenden Städte („Gedächtnis“) geht, wird dabei durch die Formel

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} & \text{für } j \in J_i^k \\ 0 & \text{für } j \notin J_i^k \end{cases}$$

beschrieben. Diese setzt das Pheromonlevel τ und die Attraktivität η eines Pfades, abhängig von den Steuerungsparametern α und β , ins Verhältnis zu Pheromonlevel und die Attraktivität aller wählbaren Pfade. Ist j nicht in der Menge der noch zu besuchenden Städte ist die Wahrscheinlichkeit, dass die Ameise diese Stadt wählt gleich Null. Die Pheromonevaporation wird durch die Formel

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

modelliert. Das Pheromonlevel zum Zeitpunkt t + 1 berechnet sich dabei als gewichtetes Pheromonlevel zum Zeitpunkt t addiert mit der Veränderung der Pheromonmenge durch Evaporation. Eine Abwandlungen dieses Algorithmus ist das Max-Min Ant System [9] bei dem nur die Ameise mit der besten Lösung Einfluss auf die Veränderung der Pheromonwerte hat und es eine explizite obere und untere Schranke für diese gibt. Ant Colony System ist dem sehr ähnlich, führt aber noch eine Kandidatenliste ein, in der die nächsten unbesuchten Städte stehen. Die nächste Stadt wird dabei nach der Formel

$$j = \begin{cases} \operatorname{argmax}_{u \in J_i^k} \{[\tau_{iu}(t)] \cdot [\eta_{iu}]^\beta\} & \text{für } q \leq q_0 \\ J & \text{für } q > q_0 \end{cases}$$

bestimmt, wobei J ähnlich wie bei Ant System ermittelt wird, nämlich nach

$$p_{ij}^k = \frac{[\tau_{ij}(t)] \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)] \cdot [\eta_{il}]^\beta},$$

q eine Zufallsvariable und q_0 ein weiterer Parameter zum Beeinflussen des Algorithmus ist. Die Wahl des weiteren Weges fällt hier, abhängig von q und q_0 , auf die nächste Stadt mit dem höchsten gewichteten Pheromonlevel oder mit der größten Wahrscheinlichkeit p_{ij}^k .

Eine weitere Abwandlung ist das Rank Based Ant System[11]. Hierbei werden die Lösungen bewertet und die Pheromonwerte abhängig von dieser Wertung verändert. Der Vorteil von Ant Colony Optimization gegenüber vielen anderen Graphenalgorithmien ist die Tatsache, dass er sich an sich ändernde Bedingungen, also dynamische Graphen, anpassen kann ohne neu beginnen zu müssen. Das macht ihn zum Beispiel für Routing und Verkehrsplanung sehr interessant.

Algorithmus 1 Ant System für das Traveling Salesman Problem nach [10]

```
1: for every edge (i,j) do
2:   // initialization
3:    $\tau_{ij} = \tau_0$ 
4: end for
5: for k = 1 to m do
6:   place ant k on a randomly chosen city
7: end for
8: let  $T^+$  be the shortest tour found from beginning and  $L^+$  its length
9: for  $t = 1$  to  $t_{max}$  do
10:  // main loop
11:  for k = 1 to m do
12:    build tour  $T^k(t)$  by applying n-1 times the following steps: choose the next
    city j with probability  $p_{ij}$  where i is the current city
13:  end for
14:  for k = 1 to m do
15:    compute length  $L^k(t)$  of tour  $T^k(t)$  produced by ant k
16:  end for
17:  if an improved tour is found then
18:    update  $T^+$  and  $L^+$ 
19:  end if
20:  for every edge (i,j) do
21:    update pheromone trails
22:  end for
23: end for
24: print the shortest tour  $T^+$  and its length  $L^+$ 
```

Bei dem *Particle Swarm Optimization* (PSO) [12] Ansatz lässt man „Partikel“ nach der Lösung für ein Optimierungsproblem suchen, und nimmt sich dabei Fische und Vögel in einem Schwarm zum Vorbild. Dazu initialisiert man die Partikel indem sie zufällig in einen mehrdimensionalen Lösungsraum verteilt werden und anhand einer Bewertungsfunktion¹¹ bewertet. Jeder Partikel kennt seinen bisher optimalen Wert sowie durch Kommunikation den seiner nächsten Nachbarn und das globale Optimum. Bei jedem Berechnungsschritt wird eine Richtung und Geschwindigkeit für jeden Partikel bestimmt. Diese hängt sowohl vom eigenen (lokalen) Optimum ab, als auch von dem der Nachbarn und dem aller Partikel (global). Dann werden sie entsprechend im Lösungsraum verschoben. Partikel werden hierbei zu optimalen Lösungen in ihrer Umgebung gezogen. Nach mehrmaligem Wiederholen bewegen sich die Partikel auf eine Lösung zu. Diese Vorgehensweise führt leider dazu, dass unter bestimmten Umständen schnell ein lokales Maximum oder Minimum gefunden wird und alle Partikel in diesem stagnieren. Um doch noch ein globales Minimum oder Maximum zu finden kann man einige Partikel neu initialisieren und so für eine höhere Diversität bei der Lösungssuche sorgen. Dieses Verfahren wird vor allem zur Optimierung genutzt und findet zum Beispiel Anwendung beim Training neuronaler Netze.

Immer mehr Ressourcen im Netz¹² werden auf Servern in Serverfarmen gehostet. Hierbei werden Server in Cluster¹³ aufgeteilt, die einen virtuellen Server bilden und alle die gleichen Anfragen bearbeiten können. Einzelne Server können anderen Clustern zugeteilt werden, was allerdings Zeit kostet um die neu benötigten Programme und Daten zu laden. Nikrani und Tovey stellen in [13] eine Methode der Ressourcenzuteilung in einer großen Serverfarm vor. Sie nutzen die Ähnlichkeit der Ressourcenverteilung bei Bienen und Internetservern. Anfragen an eine Serverfarm sind über den Tag sehr unterschiedlich verteilt und auch die Anfragenmenge unterscheidet sich von Resource zu Resource stark. So wird zum Beispiel eine Seite, die auf der Startseite von digg.com erscheint von tausenden Nutzern innerhalb kurzer Zeit besucht, während sie zuvor vielleicht nur einige hundert Besucher pro Tag zählte. Dieses Problem ist zu vergleichen mit den Blüten, die von einem Bienenvolk besucht werden. Auch hier unterscheidet sich die Nahrungsmenge von Blüte zu Blüte und ist über den Tag unterschiedlich verteilt. Nikrani und Tovey stellen eine mögliche Lösung vor. Sie stellen einen virtuellen „Tanzplatz“ zur Verfügung auf dem ein „advertisement“ vom Server platziert wird sobald eine Anfrage an ihn gestellt wird. Dieses zieht weitere Server an, die die Anfrage bearbeiten können. Je länger eine Anfrage unbeantwortet bleibt, je größer sie ist und je attraktiver sie ist in Bezug auf zu erwartenden Gewinn, umso mehr „tanzt“ sie und umso mehr zieht sie frei werdende Server an. Unter den Servern eines Clusters gibt es eine Kundschafterin und viele Sammlerinnen. Kundschafterinnen können sich jederzeit zufällig einem virtuellen Server zuteilen. Sammlerinnen teilen sich entsprechend den „advertisements“

¹¹ zum Beispiel die Distanz zu einem bestimmten Punkt

¹² Dateien, Daten in Datenbanken und mehr

¹³ Zusammenschluss mehrerer Server

zu. Auf diese Weise kann die Ressourcenverteilung sehr leicht an sich ändernde Bedingungen angepasst werden. Diese Methode der Serverallokation bringt Performancevorteile gegenüber der Methode eine Warteschlange für jedes Cluster zu nutzen und nur zu bestimmten Zeitpunkten eine Neuzuteilung von Servern zu anderen Clustern zuzulassen sofern die Anzahl und Art der Ressourcenanfragen nicht sehr homogen sind.

Swarm-Bots sind ein Ansatz Schwarmverhalten in der Robotik anzuwenden. Mehrere kleine Roboter, sogenannte s-bots [14] sind ausgerüstet mit verschiedenen Motoren, Sensoren, Kommunikationskanälen und Greifern. Ein s-bot kann also Bewegungen ausführen, Informationen aus der Umwelt wahrnehmen, mit anderen s-bots kommunizieren und Objekte oder andere s-bots greifen und wieder loslassen. Durch die Verteilung der Hardware entsteht ein sehr robustes und ausfallsicheres System. Je nach Aufgaben können sie diese eigenständig lösen oder sich zu einem größeren Swarm-Bot (nach [15] etwa 30 bis 35 s-bots) zusammenschließen indem sie sich mit Hilfe ihrer Greifer zu verschiedenen Formationen verbinden. Auf diese Weise können zum Beispiel große Hindernisse wie ein Graben überwunden (siehe Abbildung 2) oder eine Last bewegt werden, die für einen einzelnen s-bot zu schwer wäre. S-bots fahren auf Rädern und Ketten und sind damit besonders beweglich. Kommunizieren können sie mit einer eingebauten Kamera und LEDs sowie Lautsprecher und Mikrofon. Anwendung sollen Swarmbots vorallem in der Rettung finden [16]. Sie sind dafür aufgrund ihrer Robustheit, Geländegängigkeit, geringen Größe und Beweglichkeit besonders gut geeignet. Einsatzszenarien beinhalten zum Beispiel das Auskundschaften eines einsturzgefährdeten Gebäudes nach Überlebenden nach einem Erdbeben. Hier könnte ein Swarm-Bot wertvolle Erkundungsarbeit leisten und lebenswichtige Informationen für die Rettungskräfte liefern.



Abbildung 2. 2 s-bots beim Überqueren eines Grabens (www.swarm-bots.org)

3 Zusammenfassung und Zukunftsausblick

Zusammenfassend lässt sich sagen, dass Schwarmintelligenz ein sehr vielversprechender und relativ neuer Ansatz für die Informatik ist. Mit ihr lassen sich komplexe Probleme lösen mit relativ einfachen Mitteln realisieren und simulieren und ist dabei sehr robust und flexibel.

Im Bereich der Optimierung und speziell des Routings im Netz und der Logistik wird sie wohl auch in Zukunft aufgrund der bisherigen guten Ergebnisse zum Einsatz kommen.

In der Natur hat sich die Intelligenz der Masse bewährt. Bleibt die Frage, ob dies auch für den Menschen gilt. Schwarmintelligenz und Entscheidungen in einer großen Gruppe treffen funktioniert nur gut wenn die einzelnen Individuen unabhängige und unbeeinflusste Entscheidungen treffen und nicht darauf warten, dass sie gesagt bekommen was sie tun müssen. Das trifft leider allzu oft erwiesenermaßen nicht auf uns zu.

Literatur

1. von Frisch, K.: The dance language and orientation of bees. (1967)
2. Audriusa: Bee dance (waggle dance). http://en.wikipedia.org/wiki/Image:Bee_dance.png (2006) GNU FDL.
3. Partridge, B.L.: Internal dynamics and the interrelations of fish in schools. *Journal of Comparative Physiology* **114**(3) (September 1981) 313–325
4. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. *ACM Computer Graphics* **21**(4) (1987) 25–34
5. Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tâche, F., Sad, I., Durier, V., Canonge, S., Amé, J.M., Detrain, C., Correll, N., Martinoli, A., Mondada, F., Siegwart, R., Deneubourg, J.L.: Social integration of robots into groups of cockroaches to control self-organized choices. *Science* **318**(5853) (2007) 1155–1158
6. Walter, I.: Der schlaue schwarm - spektakulärer versuch mit 300 freiwilligen. Quarks & Co (2007)
7. Dorigo, M.: Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy (1992)
8. Dorigo, M., Maniezzo, V., Coloni, A.: The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics* **26**(1) (1996) 29–41
9. Stützle, T., Hoos, H.H.: Max–min ant system. *Future Generation Computer Systems* **16**(8) (2000) 889–914
10. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press (1999)
11. Bullnheimer, B., Hartl, R., Strauss, C.: A new rank based version of the ant system: A computational study. (1997)
12. Kennedy, J., Eberhart, R.C.: *Swarm intelligence*. (2001)
13. Nakrani, S., Tovey, C.: On honey bees and dynamic allocation in an internet server colony
14. Dorigo, M., Tuci, E., Mondada, F., Nolfi, S., Deneubourg, J.L., Floreano, D., Gambardella, L.: The SWARM-BOTS project. *Künstliche Intelligenz* (2005, in press)

15. Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I., Floreano, D., Deneubourg, J.L., Nolfi, S., Gambardella, L., Dorigo, M.: Swarm-bot: a new distributed robotic concept. *Autonomous Robots* **17**(2-3) (2004) 193-221
16. Mondada, F., Floreano, D., Guignard, A., Deneubourg, J.L., Gambardella, L., Nolfi, S., Dorigo, M.: Search for rescue: an application for the swarm-bot self-assembling robot concept. Technical report, LSA2 - I2S - STI, Swiss Federal Institute of Technology, Lausanne, Switzerland (September 2002)