**LECTURE NOTES**

**Selected Topics in Computational Intelligence I**

**Session 6**

**Optimization Problem**

**LEARNING OUTCOMES**

**Tujuan Instruksional Umum :**

1. Mahasiswa dapat menjelaskan contoh-contoh optimization Problem

**Tujuan Instruksional Khusus :**

1. Mahasiswa dapat menjelaskan langkah-langkah pemodelan untuk memecahkan masalah optimisasi problem.
2. Mahasiswa dapat menerapkan beberapa algoritma untuk memecahkan masalah optimisasi problem.

1. **One Max Problem**

Masalah Max One adalah masalah yang sangat sederhana di mana evolusi digunakan untuk menemukan "gen" tertentu. Sebuah gen pada dasarnya adalah bagian dari teks yang diisi dengan nilai-nilai biner acak (0 atau 1), sebuah string biner.

Contoh "gen" dari individu acak: 1001011011011101110100101010101110001

Target "gen" untuk masalah: 111111111111111111111111111111111

Tujuan dalam masalah ini adalah menemukan string biner yang identik dengan target yang ditentukan. Karena kita tidak bisa hanya mengatur string kita agar sama dengan target, kita harus menggunakan metode yang tersedia dalam algoritma evolusioner untuk "berevolusi" populasi kita sampai akhirnya target muncul. Problem ini dianggap baik sebagai tes untuk algoritma evolutionary algoritma pada tahap awal pengembangan.

Untuk menemukan kandidat yang bisa menghasilkan anak-anak kita menggunakan mekanisme seleksi orang tua, dan untuk membuat anak-anak kita menggunakan fungsi reproduksi.

Sebagai tambahan twist untuk masalah, algoritma memiliki dukungan untuk string binary manual. Jadi misalnya targetnya bisa 1001011011011 dan bukan semua 1.

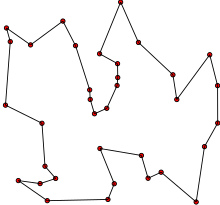
Dalam masalah sederhana ini, Fitness dihitung dengan hanya membandingkan individu yang berevolusi dengan target. Maka Fitness adalah nilai dari 0,0 hingga 1,0, yang ditemukan dengan memberi individu satu titik per nilai yang benar dalam gen, dan kemudian membaginya dengan panjang gen. Dalam contoh gen diatas , fitnessnya adalah 18/33 = 0,54.

1. **Traveling Salesman Problem**

Permasalahan TSP (Traveling Salesman Problem ) adalah permasalahan dimana seorang salesman harus mengunjungi semua kota dimana tiap kota hanya dikunjungi sekali, dan dia harus mulai dari dan kembali ke kota asal. Tujuannya adalah menentukan rute dengan jarak total atau biaya yang paling minimum. Permasalahan TSP merupakan permasalahan yang memang mudah untuk diselesaikan dengan algoritma Brute Force, tetapi hal itu hanya dapat dilakukan dengan jumlah kota atau simpul yang tidak banyak. Kompleksitas algoritma untuk permasalahan TSP dengan algoritma Brute Force adalah O(n!) dengan catatan n adalah jumlah kota atau simpul dan setiap kota atau simpul terhubung dengan semua kota atau simpul lainnya.

Dalam menyelesaikan TSP ini, setiap kota diassosiakan sebagai himpunan jawaban dari solusi yang ditawarkan. Misalan jika ada kota A- B-C-D-E, maka salah satu solusi yang ditawarkan adalah misalnya S= {A-C-E-B-D-A}. Setiap kota akan menjadi gen (A,B,C, D dan E) menjadi pembentuk solusi. Fitnes function yang sering digunakan pada masalah TSP adalah jarak yang terpendek. Jaraka dapat dihitung dengan menggunakan prinsip eucludian atau manhattan.

Problem TSP ini sering digunakan untuk menguji program dalam menyelesaikan optimasi problem khususnya masalah diskret problem. Karena masalah ini mudah untuk dipahami. Perhitungan fitnessnya juga sederhana, namun bias mewakili kerumitan dalam proses optimisasi.

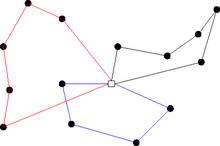


Gambar 6.1 Model TSP dengan beberapa kota

**6.3 Vehicle Routing Problem (VRP)**

Masalah VRP adalah optimasi kombinatorial dan masalah pemrograman integer yang bertanya, "Apa rangkaian rute optimal untuk armada kendaraan untuk dilalui guna mengantarkan ke satu set pelanggan tertentu?". Ini generalisasi masalah salesman perjalanan terkenal (TSP). Ini pertama kali muncul dalam sebuah makalah oleh George Dantzig dan John Ramser pada tahun 1959, di mana pendekatan algoritmik pertama ditulis dan diterapkan pada pengiriman bensin. Seringkali, konteksnya adalah pengiriman barang yang ditempatkan di depot pusat kepada pelanggan yang telah memesan barang-barang tersebut. Tujuan dari VRP adalah untuk meminimalkan total biaya rute. Pada tahun 1964, Clarke dan Wright memperbaiki pendekatan Dantzig dan Ramser menggunakan pendekatan greedy yang efektif yang disebut algoritma tabungan.

Menentukan solusi optimal untuk VRP adalah NP-hard, [2] sehingga ukuran masalah yang dapat dipecahkan, optimal, menggunakan pemrograman matematika atau optimasi kombinatorial mungkin terbatas. Oleh karena itu, pemecah komersial cenderung menggunakan heuristik karena ukuran dan frekuensi VRP dunia nyata yang harus dipecahkan. VRP memiliki banyak aplikasi yang jelas dalam omputer. Bahkan, penggunaan program pengoptimalan omputer dapat memberikan penghematan sebesar 5% untuk sebuah perusahaan.

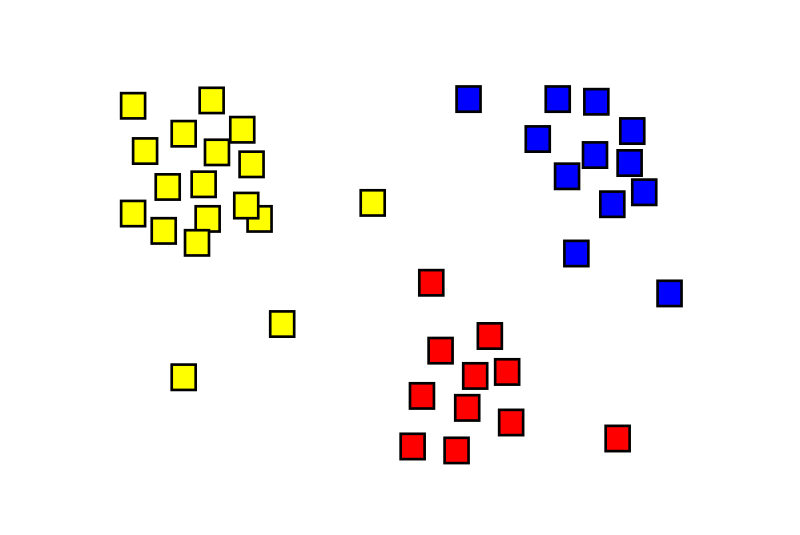


Gambar 6.2 Model VRP dengan beberapa kota

6.4 Clustering Problem

Analisis kluster atau pengelompokan adalah tugas mengelompokkan seperangkat objek sedemikian rupa sehingga objek dalam kelompok yang sama (disebut kluster) lebih mirip (dalam arti tertentu) satu sama lain daripada dengan yang ada di kelompok lain (kluster). Ini adalah tugas utama penambangan data eksplorasi, dan teknik umum untuk analisis data statistik, yang digunakan dalam berbagai bidang, termasuk pembelajaran mesin, pengenalan pola, analisis gambar, pengambilan informasi, bioinformatika, kompresi data, dan grafik komputer.

Analisis kluster itu sendiri bukanlah satu algoritma spesifik, tetapi tugas umum yang harus dipecahkan. Ini dapat dicapai oleh berbagai algoritma yang berbeda secara signifikan dalam pemahaman mereka tentang apa yang membentuk suatu kelompok dan bagaimana menemukan mereka secara efisien. Gagasan populer gugus termasuk kelompok dengan jarak kecil antara anggota klaster, area padat dari ruang data, interval atau distribusi statistik tertentu. Clustering karenanya dapat diformulasikan sebagai masalah optimisasi multi-tujuan. Algoritma pengelompokan dan pengaturan parameter yang tepat (termasuk parameter seperti fungsi jarak yang digunakan, ambang batas kepadatan atau jumlah kelompok yang diharapkan) bergantung pada kumpulan data individual dan penggunaan yang dimaksudkan dari hasil. Analisis cluster seperti itu bukan tugas otomatis, tetapi proses berulang penemuan pengetahuan atau interaktif multi-tujuan optimasi yang melibatkan percobaan dan kegagalan. Seringkali perlu untuk memodifikasi data preprocessing dan parameter model sampai hasilnya mencapai properti yang diinginkan.



Gambar 6.3 Ilustrasi Clustering menjadi 3 bagian

**SIMPULAN**

Banyak persoalan optimisasi yang mudah dimodelkan dengan memahami model dasar problem yang sudah ada. Algoritma-algoritma metahueristik dapat diuji performancenya dengan mencoba menyelesaikan persoalan optimasi tersebut.

**DAFTAR PUSTAKA**

* Adries P. Engelbrect. (2007), ***Computational Intelligence An Introduction***. 2nd ed. John Wiley & Sons. USA.
* Bansal, Jagdish Chand, Pramod Kumar Singh, and Nikhil R. Pal . (2017) "***Evolutionary and Swarm Intelligence Algorithms***", Springer,